

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-251289

(43)Date of publication of application : 14.09.2001

(51)Int.Cl.

**H04L 9/08**

**G09C 1/00**

**G09C 1/04**

(21)Application number : 2000-059091

(71)Applicant : NEC CORP

(22)Date of filing : 03.03.2000

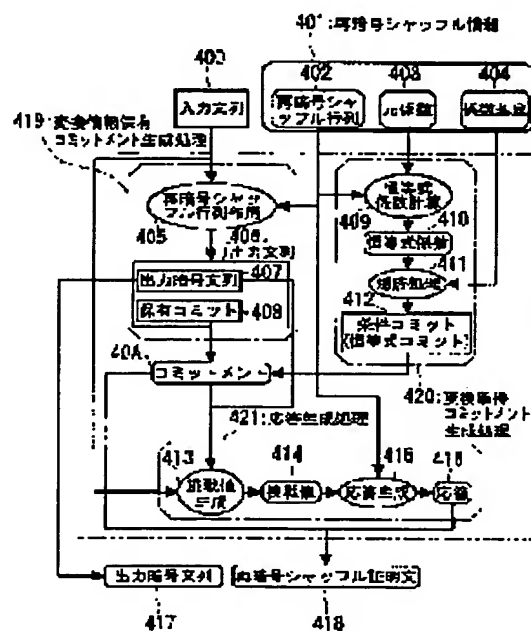
(72)Inventor : FURUKAWA JUN

(54) METHOD AND DEVICE FOR AUTHENTICATED RE-ENCIPHERMENT SHUFFLE, METHOD AND DEVICE FOR VERIFYING RE- ENCIPHERMENT SHUFFLE AND METHOD, AND DEVICE FOR GENERATING INPUT SENTENCE STRING AND RECORDING MEDIUM

**(57)Abstract:**

**PROBLEM TO BE SOLVED:** To provide certified re-encipherment shuffle having a certifying method and a verifying method in which calculation quantity can be made proportional to the number of input cipher texts and the calculation quantity can be made small, and to provided a method for verifying this certified re-encipherment shuffle.

**SOLUTION:** Re-encipherment is expressed as a kind of general conversion, and the authentication of re-encipherment shuffle can be constituted of the a



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2001-251289  
(P2001-251289A)

(43) 公開日 平成13年9月14日 (2001.9.14)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード*(参考)
H 0 4 L 9/08		G 0 9 C 1/00	6 2 0 A 5 J 1 0 4
G 0 9 C 1/00	6 2 0		6 4 0 Z 9 A 0 0 1
	6 4 0	1/04	
1/04		H 0 4 L 9/00	6 0 1 C

審査請求 有 請求項の数54 O L (全 83 頁)

(21) 出願番号 特願2000-59091(P2000-59091)

(22) 出願日 平成12年3月3日 (2000.3.3)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 古川 潤

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100080816

弁理士 加藤 朝道

Fターム(参考) 5J104 AA01 AA18 EA16 JA23 NA09

NA12 NA29 NA32

9A001 BB02 BB03 BB04 EE03 FF01

GG01 GG05 GG22 JJ18 KK56

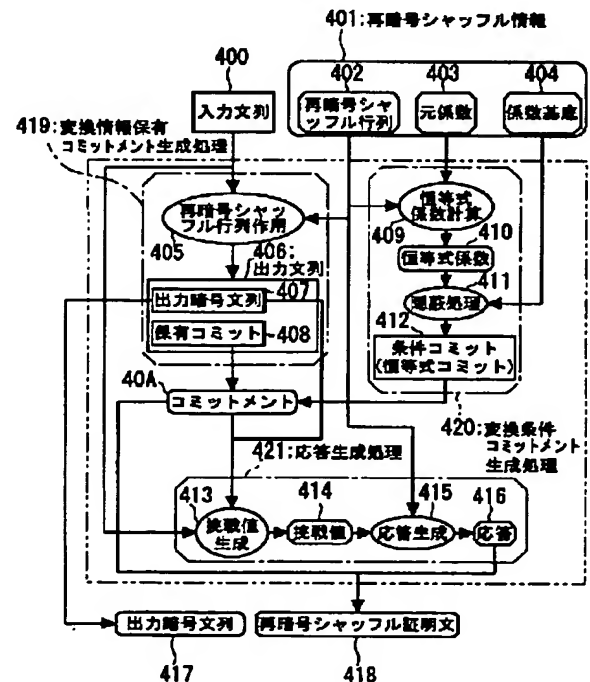
LL03

(54) 【発明の名称】 証明付再暗号シャッフル方法と装置、再暗号シャッフル検証方法と装置、入力文列生成方法と装置及び記録媒体

(57) 【要約】

【課題】 計算量が入力暗号文の数に比例しかつその計算量が少ない証明生成および検証方法をもつ証明付再暗号シャッフルおよびその検証方法の考案。

【解決手段】 再暗号シャッフルをより一般的な変換の一種として表現し、この変換情報を保有していることの証明と変換の満たす条件の証明の二つを合わせて再暗号シャッフルの証明を構成するものであり、二種の証明は、それぞれ入力暗号文数に依存せず短く、変換情報保有の証明は、挑戦値から上記変換に依存して応答を生成するため、応答と挑戦値の関係に変換の満たす条件が反映され、挑戦値に依存しない応答と挑戦値の関係式が存在し、これが成り立つことから変換の満たす条件を証明する。証明すべき条件として、再暗号シャッフルに対応する変換の満たす条件を選べば両証明をもってして再暗号シャッフルの証明を構成できる。



## 【特許請求の範囲】

【請求項1】複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から前記出力暗号文列への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成ステップと、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成ステップと、再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項2】入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を出力する再暗号シャッフル検証方法において、前記入力文列と、前記出力暗号文列と、前記入力文列から前記出力暗号文列への変換情報の保有に関する変換情報保有コミットメントと、応答と、挑戦値とから、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、前記変換の満たす条件に関する変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、

を含み、前記変換情報保有検証ステップと前記変換条件検証ステップの検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証方法。

【請求項3】請求項1に記載の証明付再暗号シャッフル方法において、前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再

暗号シャッフル情報から積和演算のみを用いて生成し、前記恒等式の係数、または前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとし、

前記応答生成ステップは、前記応答を、再暗号シャッフル情報と挑戦値から積和演算のみで生成し、

前記表現は、表現値と基底を対応付けるものであって、無作為に与えられた表現値と基底から、これらを対応付ける表現を計算することは、計算量的に困難とされており、

前記挑戦値は、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分であり、

前記挑戦値生成関数は、与えられた入力から複数の成分を出力するものであって、それらの出力からは入力を求めること、出力成分間の関係を意図して入力を決定することが、計算量的に困難である関数である、

20 ことを特徴とする証明付再暗号シャッフル方法。

【請求項4】請求項2に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、ことを特徴とする再暗号シャッフル検証方法。

【請求項5】請求項3に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、前記準応答は、再暗号シャッフル検証における、変換情報保有検証処理には使用されないものであり、応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、

前記応答生成ステップは、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、

50 前記再暗号シャッフル証明文は、前記変換情報保有コミ

ットメントと前記変換条件コミットメントと前記応答と前記準応答とを含む、

ことを特徴とする証明付再暗号シャッフル方法。

【請求項6】請求項4に記載の再暗号シャッフル検証方法において、

前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証方法。

【請求項7】請求項1に記載の証明付再暗号シャッフル方法において、

前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、

前記変換条件コミットメント生成ステップは、複数の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成ステップと、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成ステップと、の両ステップの一方または両方を複数含み、

前記応答生成ステップは、前記応答と、前記変換条件コミットメント生成処理に応じて複数の準応答を生成し、

前記再暗号シャッフル証明文は、複数の前記変換条件コミットメントと、このコミットメントに対応する準応答と、前記応答と、前記変換情報保有コミットメントとを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項8】請求項2に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、複数の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証ステップと、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証ステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。

【請求項9】請求項1に記載の証明付再暗号シャッフル方法において、

前記変換情報保有コミットメント生成ステップは、複数の変換情報保有コミットメント生成ステップよりなり、前記各変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、2番目以降の変換情報保有コミットメント生成処理は、1番目の変換情報保有コミットメント生成ステップと共通する出力の生成を省略し、

前記変換条件コミットメント生成ステップは、複数の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとするステップと、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成するステップと、の両ステップの一方または両方を複数含み、

前記応答生成ステップは、前記変換情報保有コミットメント生成ステップに応じて複数の応答を生成し、前記変換条件コミットメント生成ステップに応じて複数の準応答を生成し、

前記再暗号シャッフル証明文は、前記複数の応答と複数の知識のコミットメントと複数の変換条件コミットメントとそれに対応する準応答とよりなるものであ



る、ことを特徴とする証明付再暗号シャッフル方法。

【請求項10】請求項2に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、複数個の変換情報保有検証ステップよりなり、前記各変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、複数個の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証するステップと、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証するステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。

【請求項11】請求項3または請求項5に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル方法。

【請求項12】請求項3または請求項5に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものであることを特徴とする証明付再暗号シャッフル方法。

【請求項13】請求項7または請求項9に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含む、ことを特徴とする証明付再暗号シ

ャッフル方法。

【請求項14】証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数、または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する、ことを特徴とする入力文列生成方法。

【請求項15】請求項14に記載の入力文列生成方法において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする、ことを特徴とする入力文列生成方法。

【請求項16】請求項14に記載の入力文列生成方法において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成方法であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成方法。

【請求項17】請求項14に記載の入力文列生成方法において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する、ことを特徴とする入力文列生成方法。

【請求項18】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する、ことを特徴とする証明付公開鍵列生成方法。

【請求項19】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、

前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、

を備え、

前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文と

して出力する、ことを特徴とする証明付再暗号シャッフル装置。

【請求項 20】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に  
入力される前記入力文列と、  
前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、  
前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満たす条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、  
を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、  
前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換情報を保有して  
いることを検証する、変換情報保有検証部と、  
前記変換条件コミットメントと、前記応答と、前記挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証部と、  
を備え、  
前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証装置。

【請求項 21】請求項 19 に記載の証明付再暗号シャッフル装置において、  
前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成する手段を備え、  
前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力する手段を備え、  
前記応答生成部が、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する手段を備えたことを特徴とする証明付再暗号

シャッフル装置。

【請求項 22】請求項 20 に記載の再暗号シャッフル検証装置において、  
前記変換情報保有検証部が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証する手段を備え、  
前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する手段を備えた、ことを特徴とする再暗号シャッフル検証装置。

【請求項 23】請求項 21 に記載の証明付再暗号シャッフル装置において、  
前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成する手段と、  
前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する手段と、を備え、  
前記準応答は、これは応答と挑戦値との多項式であり、  
前記多項式の係数が準元係数であり、  
前記応答生成部は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成する手段を備え、  
前記再暗号シャッフル証明文が、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答よりなる、  
ことを特徴とする証明付再暗号シャッフル装置。

【請求項 24】請求項 22 に記載の再暗号シャッフル検証装置において、  
前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証装置。

【請求項 25】請求項 19 に記載の証明付再暗号シャッフル装置において、  
前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成する手段を備え、  
前記変換条件コミットメント生成部を複数備え、前記入

10

20

30

40

50

力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成部と、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を含み、

前記応答生成部が、前記応答と、前記複数の変換条件コミットメント生成部に依拠して複数の準応答を生成する手段を備え、  
前記再暗号シャッフル証明文が、複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントよりなる、ことを特徴とする証明付再暗号シャッフル装置。

【請求項 26】請求項 20 に記載の再暗号シャッフル検証装置において、

前記変換情報保有検証部が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証部が、複数の変換条件検証部よりなり、

前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証部と、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。

【請求項 27】請求項 19 に記載の証明付再暗号シャッフル装置において、

前記変換情報保有コミットメント生成部が、複数の変換情報保有コミットメント生成部よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、2 番目以降の変換情報保有コミットメント生

成処理は 1 番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、

前記変換条件コミットメント生成部が、複数の変換条件コミットメント生成部よりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成部と、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を複数含む、

20 前記応答生成処理が、前記複数の変換情報保有コミットメント生成部の出力に依拠して複数の準応答を生成し、前記複数の変換条件コミットメント生成部の出力に依拠して複数の準応答を生成する手段を備え、

前記再暗号シャッフル証明文は、前記複数の準応答と複数の知識のコミットメントと複数の変換条件コミットメントとそれに対応する準応答よりなる、ことを特徴とする証明付再暗号シャッフル装置。

【請求項 28】請求項 20 に記載の再暗号シャッフル検証装置において、

30 前記変換情報保有検証部が、複数の変換情報保有検証部よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証部が、複数の変換条件検証部よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証部と、

40 前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。

【請求項 29】請求項 21 または請求項 23 に記載の証明付再暗号シャッフル装置において、

前記変換条件コミットメント生成部における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル装置。

【請求項30】請求項21または請求項23に記載の証明付再暗号シャッフル装置において、前記変換条件コミットメント生成部における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものである、ことを特徴とする証明付再暗号シャッフル装置。

【請求項31】請求項25または請求項27に記載の証明付再暗号シャッフル装置において、前記変換条件コミットメント生成部における複数の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものを含む、ことを特徴とする再暗号シャッフル装置。

【請求項32】請求項19記載の前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成装置。

【請求項33】請求項32に記載の入力文列生成装置において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成装置。

【請求項34】請求項32に記載の入力文列生成装置において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成装置であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成装置。

【請求項35】請求項32に記載の入力文列生成装置において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成装置。

【請求項36】与えられた入力から一意的に決定される疑

似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する手段を備えたことを特徴とする証明付公開鍵列生成装置。

【請求項37】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する再暗号シャッフル装置であって、

(a) 前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成処理と、

(b) 前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成処理と、

(c) 再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、

(d) 前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、

の前記(a)乃至(d)の処理を証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項38】入力文列と、証明付再暗号シャッフル装置から出力される出力暗号文列と、再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満たす条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、

(a) 前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、

(b) 前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、

(c) 前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外是不受理を出力する処理、

の前記(a)乃至(c)の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 39】請求項 37 に記載の記憶媒体において、前記変換情報保有コミットメント生成処理が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成し、前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力し、前記応答生成処理は、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する、前記各処理を前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 40】請求項 38 に記載の記憶媒体において、前記変換情報保有検証処理は、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、前記変換条件検証処理は、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、前記各処理を前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 41】請求項 37 に記載の記憶媒体において、前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、前記応答生成処理は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、

前記再暗号シャッフル証明文として、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答を出力する、前記各処理を前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 42】請求項 40 に記載の記録媒体において、前記変換条件検証処理が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、前記変換条件検証処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 43】請求項 37 に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、複数の前記変換条件コミットメント生成処理を備え、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成処理と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、をからなり、

前記応答生成処理は、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数の準応答を生成し、前記再暗号シャッフル証明文として複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントを出力する、前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 44】請求項 38 に記載の記録媒体において、前記変換情報保有検証処理が、前記出力暗号文列および



変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証処理が、複数の変換条件検証処理よりなり、

前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証処理と、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証処理と、からなり、

前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項45】請求項37に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、複数の変換情報保有コミットメント生成処理よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、2番目以降の変換情報保有コミットメント生成処理は1番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、

前記変換条件コミットメント生成処理が、複数の変換条件コミットメント生成処理よりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成処理と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、を含み、

前記応答生成処理は、前記複数の変換情報保有コミットメント生成部の出力に応じて複数個の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数個の準応答を生成し、

前記再暗号シャッフル証明文として前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミ

ットメントとそれに対応する準応答とを出力する、前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項46】請求項38に記載の記録媒体において、前記変換情報保有検証処理が、複数の変換情報保有検証処理よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証処理が、複数の変換条件検証処理よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証処理と、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証処理と、を含み、

前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項47】請求項39または請求項41に記載の記録媒体において、

前記変換条件コミットメント生成処理における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しており、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項48】請求項39または請求項41に記載の記録媒体において、

前記変換条件コミットメント生成処理における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しており、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項49】請求項43または請求項45に記載の記録媒体において、

前記変換条件コミットメント生成処理における複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前

10

20

30

40

50

記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含み、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項50】請求項37に記載の記録媒体において、前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項51】請求項50に記載の記録媒体において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項52】請求項50に記載の記録媒体において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成処理であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵であり、前記各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数とする、入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項53】請求項50に記載の記録媒体において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項54】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する証明付公開鍵列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、匿名通信路の構成などに使われる、入出力暗号文の一对一の対応関係を秘匿しつつ一对一の対応の存在を保証する再暗号シャッ

ル技術および再暗号シャッフル検証技術に関する。

【0002】

【従来の技術】【従来の技術(1)】従来の証明付再暗号シャッフルの技術について、例えば特開平08-263575号公報(「文献1」という)の記載が参照される。図1に、該文献1に記載される構成を示す。なお、本願添付図面中で、合流する矢印は、矢印の元の情報が、全て集まって矢印の先へ送られることを意味し、分岐する矢印は矢印の元の情報全てまたは一部が、それぞれの矢印の先へ送られることを意味する。また破線は使用する入力文列生成法に依存することを示す。

【0003】図1において、160個ある疑似出力暗号文列は零知識証明のコミットメントである。挑戦値は、入出力暗号文とコミットメントより生成して、応答は挑戦値のビット値に応じて実線または点線の矢印で示される入力暗号文列または出力暗号文列から疑似出力暗号文列への写像の明示である。

【0004】図1に示すように、複数のElgamal入力暗号100の順序を並び替えて再び暗号化して出力する手法が紹介されている。このような処理を、「暗号シャッフル」という。この処理が正当であることを保証する為に同文献では以下の手法が紹介されている。並べ替えと再暗号の秘密乱数を毎回異なるものにして、再暗号シャッフルと同様の操作を安全変数(約160)の回数繰り返して疑似出力暗号文列を出力し、これを正当性証明のコミットメントとする。そしてこれら入出力暗号文とコミットメントのハッシュ値を挑戦値105として生成する。

【0005】この挑戦値のビット列を上から順に読み、ビットが“1”の時は入力暗号文列から、“0”の時は出力暗号文列からの並び替え(並び替えを表す写像)と再暗号化(再び暗号化した時に使ったの乱数)の明示を応答106とする。

【0006】以上のコミットメント、挑戦値、応答を再暗号シャッフルの証明文として出力する。以上においてハッシュ値のビット値に応じて対応関係を明示する方法をCut&Choose(カット・アンド・チューズ)法と言う。

【0007】【従来の技術(2)】他の従来技術としては、阿部が、1999年電子情報通信学会情報セキュリティ技術報告書で発表した、「AMix-network on Permutation Networks」(「文献2」という)が参照される。この文献2では、例えば図2に示すように、一対の入力暗号文の置換200を繰り返して、全体として複数の入力暗号文の並び替えを実現する。各入力暗号文の置換の証明を、Cut&Choose法でない方法で構成することにより、ある数より小さい入力暗号文数の証明付再暗号シャッフルとしては効率の向上を達成している。すなわち、個々の入力暗号文を置換することによって入力暗号文列全体の並び替えを実現している。個々の置換の証明は効率の良いものであるが、置換を多くそろえる必要がある。

【0008】

【発明が解決しようとする課題】しかしながら、上記した従来の技術は下記記載の問題点を有している。

【0009】従来の技術(1)においては、コミットメントの生成のために安全変数(約160)の回数暗号シャッフルしなければならない。一回の再暗号シャッフルは、入力暗号文の数の2倍の冪乗剰余演算を行わねばならず、計算量が多い。

【0010】また検証は、コミットメントを生成するのと同数の冪乗剰余演算を行わねばならず、計算量が多い。

【0011】次に従来の技術(2)においては、一対の入力暗号文の置換とその証明のコミットメントは、合わせて、8回の冪乗剰余演算が必要である。

【0012】この一置換当りの計算量は、従来の技術(1)の2入力暗号文あたりの計算量(=320)と比較すると小さいものの、入力暗号文全体のどのような並び替えでも実現できる回数にわたる一対の入力暗号文の置換が必要とされており、この数は、入力暗号文の数を $n$ とすると、 $n \log n - n + 1$ である。

【0013】このため、入力暗号文の数が増大すると、計算量が多くなる。

【0014】したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、入力暗号文数に依存せずに証明の計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。

【0015】本発明の他の目的は、検証処理を、証明同様に計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。これ以外の本発明の目的、特徴、利点等は以下の実施の形態の記載から、当業者には直ちに明らかとされるであろう。

【0016】

【課題を解決するための手段】前記目的を達成する本発明による証明付再暗号シャッフル方法は、複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント(「変換情報保有コミットメント」という)を生成する変換情報保有コミットメント生成ステップと、前記変換の満たす条件に関するコミットメント(「変換条件コミットメント」という)を生成する、変換条件コミットメント生成ステップと、再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に

用いた変数と、乱数とからなる、ことを特徴とする。

【0017】また本発明に係る再暗号シャッフル検証方法は、入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を出力する再暗号シャッフル検証方法において、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、を含み、前記変換情報保有検証処理と前記変換条件検証処理の検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする。

【0018】本発明の証明付再暗号シャッフル装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント(「変換情報保有コミットメント」という)を生成する変換情報保有コミットメント生成部と、前記変換の満たす条件に関するコミットメント(「変換条件コミットメント」という)を生成する、変換条件コミットメント生成部と、再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、を備え、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する。

【0019】本発明の再暗号シャッフル検証装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、前記再暗号シャッフル装置から出力される前記再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗

10

20

30

40

50



号文列への変換の満たす条件を検証する、変換条件検証部と、を備え、前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外是不受理を出力する。

【0020】また本発明に係る入力文列生成方法は、証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する。本発明においては、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列としてもよい。

【0021】本発明に係る記録媒体は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、

(a) 前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント(「変換情報保有コミットメント」という)を生成する変換情報保有コミットメント生成処理と、(b) 前記変換の満たす条件に関するコミットメント(「変換条件コミットメント」という)を生成する、変換条件コミットメント生成処理と、(c) 再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、(d) 前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、の前記(a)乃至(d)の処理を再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録している。

【0022】本発明に係る記録媒体は、入力文列と、前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換の満たす条件に関する変換条件コミットメントと、前記応答とからなる再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、(a) 前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、(b) 前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、(c) 前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外是不受理を出力する処理、の前記(a)乃至

(c)の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録している。

【0023】[発明の概要] 本発明は、再暗号シャッフルを、より一般的な変換の一種として表現し、この変換の情報を保有していることの証明と、この変換の満たす条件の証明との二つを合わせて、再暗号シャッフルの証明を構成している。

【0024】これら二種の証明それぞれは、従来の再暗号シャッフルの証明よりも単純であり、入力暗号文数に依存せずに、証明の計算量が短縮されたものであり、また、これらを合わせた再暗号シャッフルの証明でも、この優位性が保たれる。

【0025】変換の情報を保有していることの証明は、出力暗号文列と変換情報保有コミットメントを生成した後、挑戦値から上記変換と、変換情報保有コミットメント生成に使用する乱数とに依存して、応答を生成することにより行う。

【0026】ここで、応答と挑戦値の関係に変換が反映されるため、変換の満たす条件から挑戦値に依存せずに、応答と挑戦値の満たす関係式が存在する。この関係式をコミットして変換の満たす条件を証明する。

【0027】証明すべき変換の満たす条件として、再暗号シャッフルを表わす変換の満たす条件を選べば、両証明をもってして、再暗号シャッフルの証明を構成することができる。

【0028】

【発明の実施の形態】本発明の上記および他の目的、特徴および利点を明確にすべく、以下添付した図面を参照しながら本発明の実施の形態につき詳細に説明する。

【0029】最初に前提となる事柄について述べる。本発明において用いられる暗号方法は、確率暗号である公開鍵暗号系に属する暗号方法である。例えば、Elgamal暗号、楕円暗号、代数曲線暗号などもこれに含まれる。

【0030】本発明に係る証明付再暗号シャッフル方法は、入力暗号文の作成者全員が証明付再暗号シャッフルを行う証明者に対して、入力暗号文の制作に用いた秘密変数を漏らさなければ、証明者は再暗号シャッフル証明文を偽造できないものである。ただし、本発明に係る入力文列生成方法を併せて用いることで、入力暗号文の作成者が証明者と共謀した場合においても、証明文の偽造を防止する、ことができる。

【0031】本発明に係る証明付再暗号シャッフル方法は、変換情報保有コミットメントを生成する変換情報保有コミットメント生成処理と、変換条件コミットメントを生成する変換条件コミットメント生成処理と、応答や準応答を生成する応答生成処理とよりなり、証明文はこの上記3種類の処理より生成されるコミットメントと応答(応答および準応答)とよりなる。

【0032】本発明に係る再暗号シャッフル検証方法

10

20

30

40

50

は、入力文列と出力暗号文列と変換情報保有コミットメントと応答とから変換の情報を保有していることを検証する変換情報保有検証処理と、変換条件コミットメントと応答と準応答とから変換の満たす条件を検証する変換条件検証処理とよりなる。

【0033】[変換情報保有コミットメント生成処理] 証明付再暗号シャッフル方法を構成する変換情報保有コミットメント生成処理について説明する。

【0034】変換情報保有コミットメント生成処理は、入力文列から再暗号シャッフルに対応する変換を行って出力暗号文列を生成し、入力文列から乱数による一般の変換を行って変換情報保有コミットメントを生成する。

【0035】また入力文列に入力暗号文列と公開鍵以外が含まれている場合には、それから再暗号シャッフルに対応する変換を行ったものも変換情報保有コミットメントとする。

【0036】応答を複数個生成する場合には、異なる乱数による一般の変換を、複数個行い、変換情報保有コミットメントを、複数組生成する。

【0037】例えば、この変換を、出力暗号文列および変換情報保有コミットメントを、入力文列を基底として、再暗号化に用いた変数および乱数および並び替えに対応する値とを表現とした表現値として生成できる。

【0038】また、この表現とは、基底とある表現値を対応付けるものであり、かつ、基底と表現値から表現を計算することが計算量的に困難となる方法である必要があり、この表現方法に、冪乗剰余を用いることができる。

【0039】例えば、入力暗号文列を、 $g[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を、 $g[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ 、それ以外の入力文列の成分を、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$ 、一般の変換に対応する乱数(以下「情報隠蔽因子」と呼ぶ)を、 $A[\mu, j]; \mu=1, \dots, n+m, j=n+1, \dots, n+m'$ 、再暗号化の変数を、 $A[i, j]; i=n+1, \dots, n+m, j=1, \dots, n$ 、並び替えに対応する変換を表す変数を、 $A[i, j]; i, j=1, \dots, n$ として、出力暗号文列  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=1, \dots, l$  を、
$$g'[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A_{j,i}^{(1,1)}} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A_{j,i}^{(2,1)}} / F_p \quad i=1, \dots, n \quad \Gamma=1, \dots, l$$

と生成し、変換情報保有コミットメントを、
$$g'[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A_{j,i}^{(1,1)}} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A_{j,i}^{(2,1)}} / F_p \quad i=n+1, \dots, n+m' \quad \Gamma=1, \dots, l$$

と生成し、入力文列に  $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$  が含まれた場合の変換情報保有コミットメントを、

$$g'[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A_{j,i}^{(1,1)}} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A_{j,i}^{(2,1)}} / F_p \quad i=1, \dots, n+m' \quad \Gamma=1+1, \dots, l'$$

と生成できる。

【0040】また、これらをまとめて

$$g'[i, \Gamma] = \prod_{j=1}^{n+m'} g[j, \Gamma]^{A_{j,i}^{(1,1)}} / F_p \quad i=1, \dots, n+m'$$

10

20

30

40

50

$\Gamma=1, \dots, l'$

と記述できる。

【0041】この時、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l$  を「出力文列」と呼ぶ。ここで、 $g'[\mu, \Gamma]$  が表現値で、 $A[\mu, v]$  が表現で、 $g[\mu, \Gamma]$  が基底である。

【0042】変換情報保有コミットメントを応答の数に応じて、複数組生成する場合には、異なる  $A[\mu, j]; \mu=1, \dots, n+m, j=n+1, \dots, n+m'$  を複数用意して生成する。

【0043】この変換情報保有コミットメントと、入力文列と出力暗号文列および挑戦値に対応する応答を証明者が検証式を満たすように生成できることが、入力文列から出力暗号文列への変換の知識を有していることの証明となる。

【0044】[変換条件コミットメント生成処理] 証明付再暗号シャッフル方法を構成する変換条件コミットメント生成処理について説明する。

【0045】応答と挑戦値の関係に、入力文列から出力文列および変換情報保有コミットメントへの変換の満たす条件が反映される。そのため、挑戦値に依存せずに成り立つ、応答と挑戦値の関係式が存在する。この関係式をこの変換の満たす条件を表現するものとしてコミットしたものを変換条件コミットメントとする。

【0046】応答を複数個生成する場合は、知識の隠蔽因子の違いを関係式に反映させる。例えばこの関係式を、応答と挑戦値の多項式である恒等式とし、その係数をコミットするか、あるいはこの多項式の一部の項を準応答として、準応答の係数をコミットして変換条件コミットメントとできる。また挑戦値決定後に応答と準応答を生成すればよい。

【0047】応答の各成分は挑戦値の多項式であるが、一部のこの多項式の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものや、上記にて各3乗の和とが、挑戦値によらずに等しくなる関係を内包している恒等式を実施例では用いている。

【0048】これに対応する実施例における恒等式は、挑戦値  $c[i]$ 、応答  $r[i]$  を用いて、
$$\sum_{i=1}^n (\sum_{j=1}^n A[i, j] c[j])^2 = \sum_{i=1}^n c[i]^2 / F_q$$

や、
$$\sum_{i=1}^n (\sum_{j=1}^n A[i, j] c[j])^3 = \sum_{i=1}^n c[i]^3 / F_q$$

の関係を内包したものを用いている。

【0049】なお、

$$\sum_{j=1}^n A[i, j] c[j] / F_q \quad i=1, \dots, n$$

は、 $r[i]$  を構成する挑戦値の多項式

$$\sum_{j=1}^{n+m'} A[i, j] c[j] / F_q \quad i=1, \dots, n$$

の一部である。

【0050】例えばこれらの関係式は、入力文列から出力暗号文列および変換情報保有コミットメントへの変換を定義している変数  $A[\mu, v]; \mu=0, \dots, n+m; v=0, \dots, n+m'$  における  $A[i, j]; i, j=0, \dots, n$  が、正規直交行列であ

ることや、準置換行列であることの性質を反映した関係式である。

【0051】「置換行列」とは、正方行列で、どの行どの列にもただ一つだけ0でない成分が存在し、その値が1である行列のことである。正規直交行列でありかつ準置換行列である行列は置換行列である。

【0052】「準置換行列」とは、上記置換行列の1である成分を、1の3乗根のいずれかで置き換えたものとする。ただし、それぞれの成分毎に異なる1の3乗根で置き換えを行っても良い。この時、置換行列に対応する\*10

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n c[i]c[i]c[i] + \sum_{i=1}^n \psi[i]c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

があげられる。

【0054】ここでは、変換の満たす条件に対応する関係式を内包するように恒等式の係数 $\rho''$ 、 $\rho'[i]$ 、 $\phi[\mu]$ 、 $\psi[i]$ を求めなければならない。

【0055】また恒等式の一部

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i] / F_q$$

を準応答として、準元係数 $\lambda[\mu]$ ； $\mu=0, \dots, n$ をコミットする場合もある。

【0056】変換条件コミットメントとして、恒等式の係数または、それをコミットしたものと、準元係数をコミットしたものを生成する。実施例では恒等式の一部を、

$$v, v^{(i)} / F_p$$

のようにコミットし、準元係数を

$$u, u^{(\mu)} / F_p \quad \mu=0, \dots, n$$

のようにコミットする。

【0057】恒等式の係数をコミットすることと、準応答を用いることには、検証者が応答とコミットメントから再暗号シャッフルを特定するための情報を減じる効果がある。

【0058】[応答生成処理] 証明付再暗号シャッフル方法を構成する応答生成処理について説明する。

【0059】応答生成処理では、変換情報保有コミットメントと変換条件コミットメントと入力文列と出力暗号文列を挑戦値生成関数に入力して挑戦値をえる。

【0060】ここで、「挑戦値生成関数」とは、出力から入力を求めることや、異なる出力成分間の関係を意図して入力を決定することが計算量的に困難である関数である。これにより挑戦値が入力とコミットメントと出力とが決定後に、証明者の意図を入れずに生成されたことが保証できる。

【0061】挑戦値生成関数を用いない場合には、検証者が、入力と出力とコミットメントとが示された後に無作為に選ぶことで挑戦値を得る。

【0062】挑戦値から、再暗号シャッフル方法と情報隠蔽因子とを反映した応答や準応答を生成する。

\*の変換は再暗号シャッフルに対応している。すなわち、この変換条件コミットメント生成処理により変換の満たす条件を証明することによって変換が再暗号シャッフルであることを証明できる効果がある。

【0053】例えば、上記関係式を内包した恒等式の例として、

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i] + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

や、

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \end{aligned}$$

【0063】応答や準応答を複数個生成する場合は、各応答は異なる情報隠蔽因子を反映させる。

【0064】例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする表現値が等しくなるような応答を生成すればよい。

【0065】例えば、応答 $r[\mu]$ ； $\mu=1, n+m$ は、挑戦値 $c[\mu]$ ； $\mu=1, \dots, n+m$ を用いて、

$$r[\mu] = \sum_{v=1}^{n+m} A[\mu, v]c[v] / F_q \quad \mu=1, \dots, n+m$$

を、準応答として、

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i] / F_q$$

を生成する。

【0066】[変換情報保有検証処理] 再暗号シャッフル検証方法を構成する変換情報保有検証処理について説明する。

【0067】入力文列と出力暗号文列と変換情報保有コミットメント間の関係を、応答と挑戦値の関係が反映していることを検証する。例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする表現値が等しくなるような応答と挑戦値の関係が存在することを確認する。

【0068】例えば、挑戦値 $c[i]$ ； $i=1, \dots, n+m$ と応答 $r[i]$ ； $i=1, \dots, n+m$ が、

$$\prod_{i=1}^{n+m} g^{(i)}[i, \Gamma]^{(i)} = \prod_{i=1}^{n+m} g[i, \Gamma]^{(i)} / F_p$$

$$\Gamma=1, \dots, l'$$

が成り立つことを確認する。

【0069】挑戦値は、証明文生成に用いた値と同じ値を用いる。これは、挑戦値生成関数を用いる場合は、挑戦値生成関数への入力が証明文と入出力に存在するので可能である。

【0070】[変換条件検証処理] 再暗号シャッフル検証方法を構成する変換条件検証処理について説明する。

【0071】変換条件コミットメントより、挑戦値と応答が変換の満たす条件を反映した関係を満たしていることを検証する。

【0072】例えば、変換の満たす条件を反映した関係を内包する恒等式に、応答と挑戦値または、応答と挑戦値と準応答を代入して、恒等式が成立することを確認している。また準応答がある場合、応答と準応答と準元係数をコミットしたものより、準応答の正当性も確認する。

【0073】例えば、変換条件コミットメントとして恒\*

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] \\ & \quad / F_q \\ & = \sum_{i=1}^n c[i]c[i]c[i] + \sum_{i=1}^n \phi[i]c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

が成り立つことを確認し、また準応答の正当性を、検証式

$$u' = u[0] \prod_{i=1}^n u[i]^{r[i]r[i]} / F_p$$

が成り立つことより確認する。

【0074】また恒等式の係数の一部がコミットされて※

$$\begin{aligned} & v \{ \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] \} / F_p \\ & = v \{ \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu] \\ & \quad r[\mu] \} / F_p \\ & = v \{ \sum_{i=1}^n c[i]c[i]c[i] + \sum_{i=1}^n \phi[i]c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] \} / F_p \end{aligned}$$

が成り立つことを確認する。上式で、記号「 $\wedge$ 」は指数演算を示す。

【0075】[入力文列生成方法] 本発明に係る証明付再暗号シャッフル方法は、応答と挑戦値の係数に、入力文列から出力暗号文列と変換情報保有コミットメントへの変換を反映する必要があった。そのためには、挑戦値が与えられたときに生成できる応答が制限されていなければならない。しかし、入力暗号文の生成情報を証明者が知っている場合この制限を破れる可能性がある。そしてこれを阻止するための方法が入力文列生成方法である。

【0076】本発明に係る入力文列生成方法は、疑似乱数を生成し、これにより入力文列を変換するか、この疑似乱数を入力文列に加えることにより、入力暗号文の生成者にさえ入力文列を決定できない入力文列を生成する。

【0077】[入力文列生成方法(1)] 疑似乱数を生成して、入力暗号文列と公開鍵にくわえて入力文列とする。この時疑似乱数を決められた入力より決定して再現性を保証する。

【0078】例えば、入力暗号文列を、 $g[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を $g[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ としたとき、決まった入力から疑似乱数を、 $(n+m) \times (l'-1)$  ;  $l'-1 \geq 1$ 個生成し、これを、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$ として、入力文列を、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l'$ とする。

【0079】[入力文列生成方法(2)] 入力暗号文列を構成する各暗号文と公開鍵を、入力暗号文列と公開鍵を入力として生成した公開鍵列を構成する各公開鍵で再暗

\* 等式の係数 $\rho''$ 、 $\rho'[\mu]$ 、 $\phi[\mu]$ 、 $\phi[i]$ に対して、挑戦値 $c[i]; i=1, \dots, n+m$ と、応答 $r[i]; i=1, \dots, n+m$ が、恒等式

$$\sum_{i=1}^n r[i]r[i] + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] = \sum_{i=1}^n c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q$$

や、恒等式

※ いる場合は、代わりに、

$$\begin{aligned} & v \{ \sum_{i=1}^n r[i]r[i] + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] \} / F_p \\ & = v \{ \sum_{i=1}^n c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] \} / F_p \end{aligned}$$

や、

号化して、これをあわせて入力文列とする。

【0080】ここで「公開鍵列」とは、入力から疑似乱数を公開鍵列を構成する公開鍵の数と同数一意的に生成し、この各乱数が各公開鍵のいずれかの要素となるようにしたものを用いる。

【0081】例えば、公開鍵列を、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l$ とし、入力暗号文列を、 $\eta[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を、 $\eta[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ としたとき、入力文列  $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l$ を、検証者にも明らかな任意の正整数 $s[i]; i=1, \dots, n+m$ を用いて、 $g[i, \Gamma] = \eta[i, \Gamma] g'[i, \Gamma]^{s[i]} / F_p$ と表す。 $s[i]$ として、例えば $s[n+m]=0, s[j]=1; j=1, \dots, n+m-1$ を選ぶ。

【0082】[入力文列生成方法(3)] 各入力平文を公開鍵列を構成する対応する各公開鍵を用いて暗号化し、この公開鍵を用いて暗号化したことの証明をする。この証明文が受理された暗号文と、公開鍵をあわせて入力文列とする。

【0083】例えば、公開鍵列を、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=0, 1$ とし、平文を、 $m[i]; i=1, \dots, n; \Gamma=0, 1$ としたとき、入力暗号文を、 $\eta[i, 0] = g'[i, 0]^{s[i]} / F_p, i=1, \dots, n$   
 $\eta[i, 1] = m[i] g'[i, 1]^{s[i]} / F_p, i=1, \dots, n$ と生成し、合わせて、 $\eta[i, 0] = g'[i, 0]^{s[i]} / F_p$ となる $s[i]$ の知識を証明することで、 $g'[i, 0]$ を使って暗号化した証明文とする。

【0084】この証明文が検証された暗号文より、入力文列を

$$g[i, \Gamma] = \eta[i, \Gamma] \quad i=1, \dots, n; \Gamma=0, 1$$

$g[i, \Gamma] = g'[i, \Gamma] \quad i=n+1, \dots, n+m; \Gamma=0, 1$   
とする。

【0085】[証明付公開鍵列生成方法] 与えられた入力から一意的に疑似乱数列を発生させ、その各乱数から与えられた手続きにより作られた値を成分に持ち、かつ同じ秘密鍵を持つ公開鍵を、各乱数に対応して複数個生成する。これと同時に、全ての公開鍵が同じ秘密鍵を持つことの証明文を生成する。

【0086】複数人で秘密鍵を分散所持している場合は、各人が各秘密鍵で分散して公開鍵列を生成した後、それらを合わせて統合した公開鍵列を生成する。

【0087】例えば、疑似乱数生成器Hash(\*)が与えられ、入力\*から出力を得る。そして出力をまた入力することを繰り返すことにより、再帰的に、疑似乱数列を生成する。この数列の各値をk乗してできた数列から0, 1を除いて得られるn+m個の値よりなる数列 $g'[i, 0]; i=1, \dots, n+m$ の各値を成分に持ち、同じ秘密鍵を持つ公開鍵 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=0, \dots, 1$ を各乱数に対応して生成する。

【0088】ここで秘密鍵を $x[\Gamma]; \Gamma=1, \dots, 1$ とする

と、公開鍵列は、

$g'[i, 0] = g'[i, 0]$

$g'[i, \Gamma] = g'[i, 0]^{x[\Gamma]} / F_p \quad i=1, \dots, n+m; \Gamma=1, \dots, 1$

と表せる。

【0089】上記公開鍵列を正しく生成したことの証明文を生成する。

【0090】秘密鍵を分散所持している場合、各自で分散秘密鍵に対応する公開鍵列を生成し、最後にそれらを合わせて秘密鍵に対する公開鍵列を生成する。

【0091】

【実施例】本発明の実施例について図面を参照して説明する。以下の実施例では、Elgamal暗号を使った例に即して説明する。図中では、略語が使用されており、例えば保有コミットとは変換情報保有コミットメント、条件コミットとは変換条件コミットメント、恒等式コミットとは恒等式係数のコミットメント、準応答コミットとは準応答の係数のコミットメント、保有処理とは変換情報保有コミットメント生成処理、条件処理とは変換条件コミットメント生成処理、応答処理とは応答生成処理、保有検証処理とは変換情報保有検証処理、条件検証処理とは変換条件検証処理のことをいう。

【0092】図3は、本発明に係る証明付再暗号シャッフル装置および再暗号シャッフル検証装置の実施例における入出力について示す図である。

【0093】図3を参照すると、本発明の一実施例においては、複数の入力暗号文322と公開鍵323からなる入力文列300と、並び替え方を決めるシャッフル行列307と、再暗号化の変数である再暗号秘密乱数305と、変換情報保有コミットメントを生成するための乱数である情報隠蔽因子306と、からなる再暗号シャッフル行列304と、恒

等式の係数の種となる元係数308と、恒等式の一部である準応答319の係数である準元係数309と、これらをコミットするための係数基底310とよりなる変換条件コミットメントを生成するための諸定数と、を含む再暗号シャッフル情報303と、が、証明付再暗号シャッフル装置312に入力され、出力暗号文列313と再暗号シャッフル証明文314が出力される。

【0094】再暗号シャッフル証明文314は、恒等式の係数、または、それをコミットしたものと準応答の係数をコミットしたものとを含む変換条件コミットメント316と、変換情報保有コミットメント315と、応答317と、準応答318とを含む。

【0095】これら入力文列300と、出力暗号文列313と、再暗号シャッフル証明文314とが、再暗号シャッフル検証装置319に入力され、受理または不受理の検証結果320が出力される。

【0096】上記証明付再暗号シャッフル装置は、証明者が入力暗号文列の生成情報を知らない時のみに、再暗号シャッフル証明文を偽造できない。この偽造をいかなる場合にも阻止するために加える方法が、入力文列生成方法であり、3種類の入力文列生成方法の実施例を挙げる。また、このうち二つの入力文列生成方法で使用される証明付公開鍵列生成方法についても説明する。

【0097】以下では、証明付再暗号シャッフル方法と、入力文列生成方法と、証明付個別公開鍵列生成方法とに共通して、前提となる事項から順に説明していく。

【0098】[Elgamal領域変数] まずElgamal領域変数について説明する。

【0099】この変数は、二つの素数 $p, q$ であり、これらは、 $p=kq+1$ なる関係を満たす。ここで、 $k$ は整数である。

【0100】[挑戦値生成関数と基底生成関数] 挑戦値生成関数と基底生成関数について説明する。これらは順に、

$\text{Hash}[\mu; \mu=0, \dots, n](*)$ ,  $\text{Hash}'[\mu; \mu=0, \dots, n](*)$   
とする。

【0101】両関数の添字についているギリシャ文字 $\mu$ は0からnまでの値をとり、引数「\*」を入力すると、それぞれn+1成分のベクトルを出力する。

【0102】挑戦値生成関数の出力は、n+1個の1, 0でないq以下の整数、基底生成関数の出力は、n+1個の1, 0でないp以下の整数で位数qの $F_p$ の元(位数p-1の乗法群の位数q部分群の元)である整数である。

【0103】また、これらの関数は入出力間や出力の異なる成分間の関係を計算量的に意図して引数を決定できない関数とする。

【0104】基底生成関数の具体的な構成方法の例としては、 $|p|$ ビットを出力するハッシュ関数Hash(\*)を一つ用意して、

Hash(\*)

を計算し、次に、この計算結果をハッシュ関数の引数に  
入力してさらに計算結果を得る。これを、繰り返すこと  
により、再帰的に、数列 $h[0], h[1], h[2], \dots$ を生成し、  
その各数値を $k$ 乗をした数列 $h[0]^k, h[1]^k, h[2]^k, \dots$ を求  
める。この中から順に、1, 0でないものを $n+1$ 個選んでい  
く。

【0105】挑戦値生成関数の場合は、 $|q|$ ビットを出  
力するハッシュ関数を用いて数列を求め、その中から順  
に1, 0でないものを $n+1$ 個選んでいく(この場合、 $k$ 乗する  
操作は必要無い)。

【公開鍵】公開鍵について説明する。公開鍵は、二つの  
数値 $\eta[0, 0], \eta[0, 1]$ であり $\eta[0, 0]$ は位数 $q$ の $F_p$ の元と  
する。 $\eta[0, 1]$ は秘密鍵 $x$ を用いて、  
 $\eta[0, 1] = \eta[0, 0]^x / F_p$   
と計算される。

【0106】【入力暗号文】入力暗号文について説明す  
る。平文を、 $p$ 以下で位数 $q$ の $F_p$ の元から選び、これを $M$   
とする。これから疑似乱数生成器で生成した秘密乱数 $r$   
を用いて、入力暗号文を、  
 $(\eta[0, 0]^r, M \cdot \eta[0, 1]^r) / F_p$   
と計算する。

【0107】【再暗号化】再暗号化について説明する。  
Elgamal暗号文 $(\eta[0, 0]^r, M \cdot \eta[0, 1]^r) / F_p$ が与えられた  
時、任意の乱数 $s$ を選んで、  
 $(\eta[0, 0]^r, M \cdot \eta[0, 1]^r) \rightarrow (\eta[0, 0]^r \cdot \eta[0, 0]^s, M \cdot \eta[0, 1]^r \cdot \eta[0, 1]^s) / F_p = (\eta[0, 0]^{r+s}, M \cdot \eta[0, 1]^{r+s}) / F_p$   
なる変換を行うことを「再暗号化」という。上記変換は  
 $r$ を知らなくても実行できる。またこの変換により再暗  
号化された暗号文の復号結果はかわらない。この時の乱  
数 $s$ を、「再暗号秘密乱数」と呼ぶ。

【0108】【置換行列】置換行列について説明する。  
「置換行列」とはどの行にもまたどの列に対しても0で  
ない成分が唯一存在し、1の値をとる。ただし本実施例  
では $F_q$ 上で考える。下に例をあげる。

【0109】

0 0 0 1 0  
1 0 0 0 0  
0 1 0 0 0  
0 0 0 0 1  
0 0 1 0 0 /  $F_q$

【0110】【準置換行列】準置換行列について説明す  
る。「準置換行列」とは、置換行列の1である成分を、 $F_p$   
上の3個ある1の3乗根のいずれかで置き換えたもの  
と定義する。これらを $w, w^2, 1$ として下に準置換行列の例  
をあげる。

【0111】

0 0 0  $w^2$  0  
 $w$  0 0 0 0  
0  $w^2$  0 0 0  
0 0 0 0 1

0 0  $w$  0 0 /  $F_q$

【0112】【再暗号シャッフル】再暗号シャッフルに  
ついて説明する。入力暗号文列 $\eta[i, 0], \eta[i, 1]; i =$   
 $1, \dots, n$ の順序を入れ替えて、暗号文列 $\eta'[i, 0], \eta'[i,$   
 $1]; i = 1, \dots, n$ を生成し、さらに $n$ 個の秘密乱数 $s[i]; i =$   
 $1, \dots, n$ と、公開鍵 $\eta[0, 0], \eta[0, 1]$ を用いて、出力暗号  
文列 $g'[i, \Gamma]; i = 1, \dots, n, \Gamma = 0, 1$ を、  
 $g'[i, \Gamma] = \eta'[i, \Gamma] \cdot \eta[0, \Gamma]^{s(i)} / F_p, i = 1, \dots, n, \Gamma =$   
 $0, 1$

10 と計算する。これが、再暗号シャッフルの出力結果であ  
る。これを、「出力暗号文列」と呼ぶ。

【0113】【再暗号シャッフル行列】再暗号シャッフル  
行列について説明する。「再暗号シャッフル行列」と  
は、 $n+1$ 行 $n+1$ 列の行列で、その成分 $A[\mu, \nu]; \mu, \nu =$   
 $0, \dots, n$ が、

$A[\mu, \nu] =$

$A[i, j] \quad i, j = 1, \dots, n$  シャッフル行列307

$A[0, j] \in \mathbb{F}_p \quad j = 1, \dots, n$  再暗号秘密乱数305

$A[i, 0] \in \mathbb{F}_p \quad i = 1, \dots, n$  情報隠蔽因子306

20  $A[0, 0] \in \mathbb{F}_p$  情報隠蔽因子306

であるものである。

【0114】【再暗号シャッフル行列変換】再暗号シャ  
ッフル行列変換について説明する。これは、入力文列 $g$   
 $[\mu, \Gamma]$ に以下のように作用して、出力文列 $g'[\mu, \Gamma]$   
を出力する。

【0115】 $g'[\mu, \Gamma] = \prod_{\nu=0}^n g[\nu, \Gamma] A[\nu, \mu] / F_p$   
 $\mu = 0, \dots, n, \Gamma = 0, 1$

ここで、シャッフル行列が置換行列の場合、出力暗号文  
列を、 $g'[i, 0], g'[i, 1]; i = 1, \dots, n$ とし、展開する

30 と、ある置換 $(i, j | \pi(i) = j)$ に対して、

$g'[j, 0] = g[i, 0] \cdot \eta[0, 0]^{A[i, j]} / F_p$

$g'[j, 1] = g[i, 1] \cdot \eta[0, 1]^{A[i, j]} / F_p$

となり、これは再暗号シャッフルの出力となる。

【0116】またシャッフル行列が準置換行列の場合、  
準再暗号シャッフルの結果

$g'[j, 0] = g[i, 0]^{w^{(i)}} \cdot \eta[0, 0]^{A[i, j]} / F_p$

$g'[j, 1] = g[i, 1]^{w^{(i)}} \cdot \eta[0, 1]^{A[i, j]} / F_p$

を出力する(準再暗号シャッフルとは各出力暗号文を1ま  
たは $w$ または $w^2$ 乗すると再暗号シャッフルとなるものと  
定義する)。ここで $w[i]; i = 1, \dots, n$ は $F_q$ 上の1の3乗根の  
いずれかをとり、

【0117】【実施例(1)】本発明の一実施例をなす証  
明付再暗号シャッフル方法およびその検証方法につい  
て、図4、図5を参照して説明する。以下で、 $\Gamma = 0, 1$ を  
取るものとする。

【0118】再暗号シャッフル情報401として、再暗号  
シャッフル行列402、係数基底404、元係数403を以下の  
ように準備する。

【0119】再暗号シャッフル行列402に関しては、ま  
ず1から $n$ までの数を順に並べる。疑似乱数発生器(不

図示)を $n$ 回使って、 $n$ 個数列を発生させ、その $i$ 番目の数を $n-i+1$ で割り余りの数を求め $\pi'(i)$ とする。

【0120】 $i$ は1から $n$ 迄順に、上記並べた数の下から $\pi'(i)$ 番目の数を $\pi(i)$ とし、上記数列からこの数を取り除く作業を行い $\pi(i); i=1, \dots, n$ を決定する。シャッフル行列の第 $i$ 行目は $\pi(i)$ 列目の成分のみ値を1とし、その他を0とする。以上のようにして置換行列を生成する。

【0121】再暗号シャッフル行列のシャッフル行列以外の成分を以下のようにして生成する。まず、疑似乱数発生器により $2n+1$ 個の $F_q$ 上の数を作成し、 $A[i, 0], A[0, j], A[0, 0]; i, j=1, \dots, n$ に割り振る。以上を合わせて再暗号シャッフル行列とする。

【0122】係数基底404  $v$ 、元係数403  $r'[0]$ を生成に関しては、疑似乱数発生器で1,0でない $F_q$ 上の数を生成し、 $r'[0]$ とし、疑似乱数発生器により $/F_p$ の元を生成し $F_p$ 上でその $k$ 乗をとり1,0でないものを選び位数 $q$ の $F_p$ の元を生成し、 $v$ とする。

【0123】 $r'[0] \in_r Z_q, \neq 0, 1$

$v \in_r Z_p, \neq 0, 1, \text{ s.t. } v^q = 1 / F_p$

入力暗号文列 $\eta[i, 0], \eta[i, 1]; i=1, \dots, n$ と、公開鍵 $\eta[0, 0], \eta[0, 1]$ より、入力文列400

$g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ を、

$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$

$g[i, \Gamma] = \eta[i, \Gamma] / F_p \quad i=1, \dots, n, \Gamma=0, 1$

とする。

【0124】以下、証明付再暗号シャッフル方法を使う。

【0125】変換情報保有コミットメント生成処理419における再暗号シャッフル行列作用405により、上記暗号シャッフル行列402を入力文列400に以下の様に作用させて、出力文列406  $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ を、  
 $g'[\mu, \Gamma] = \prod_{\Gamma=0}^1 v^{-\Gamma} g[\mu, \Gamma]^{(v^{-\Gamma})} / F_p \quad \mu=0, \dots, n, \Gamma=0, 1$   
 と生成する。

【0126】ここで、 $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ を出力暗号文列407、 $g'[0, \Gamma]; \Gamma=0, 1$ を変換情報保有コミットメント408とする。

【0127】変換条件コミットメント生成処理420における恒等式係数計算409により、元係数403  $r'[0]$ と、再暗号シャッフル行列402と、を用いて、恒等式係数410  $\phi[\mu], r'[0]$ を、  
 $r'[0] = r'[0]$   
 $\phi[0] = \sum_{j=1}^n A[j, 0]A[j, 0] + r'[0]A[0, 0] / F_q$   
 $\phi[i] = 2 \sum_{j=1}^n A[j, 0]A[j, i] + r'[0]A[0, i] / F_q \quad i=1, \dots, n$   
 と生成する。

【0128】さらに、係数基底404  $v$ を用いて、隠蔽処理411により、恒等式係数410  $r'[0], \phi[0]$ を、

$$v' = v^{r'[0]} / F_p$$

$$\omega = v^{\phi[0]} / F_p$$

とコミットする。

【0129】以上より、 $\phi[i], \omega, v', v$ を変換条件コミットメント412とする。

【0130】ここで、コミットメント40 Aを、変換情報保有コミットメント408と変換条件コミットメント412とする。

【0131】応答生成処理421により、以上の入力文列400と、出力暗号文列417と、コミットメント409と、を挑戦値生成関数413の引数として、挑戦値414を、

$c[0]=1,$

$c[i] = \text{Hash}[i](g[v, 0], g[v, 1], g'[v, 0], g'[v, 1], v, \phi[v], \omega, v'; v=0, \dots, n) \quad i=1, \dots, n$

と生成し、この挑戦値から再暗号シャッフル行列02を用いて、応答416を

$r[\mu] = \sum_{v=0}^n A[\mu, v] c[v] / F_q \quad \mu=0, \dots, n$   
 と生成415する。

【0132】以上のコミットメント40 Aと応答416を、再暗号シャッフル証明文418として出力し、再暗号シャッフルの結果として出力暗号文列417を出力する。

【0133】検証方法について、図5を参照して説明する。

【0134】再暗号シャッフル検証方法により、入力文列400  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ と、出力暗号文列417  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ と、再暗号シャッフル証明文418中のコミットメント409である変換情報保有コミットメント408  $g'[0, \Gamma]; \Gamma=0, 1$ と、変換条件コミットメント412  $\phi[v], \omega, v', v; v=0, \dots, n; \Gamma=0, 1$ と、を挑戦値生成関数500に代入して、挑戦値501を、  
 $c[0]=1$   
 $c[i] = \text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], \phi[v], \omega, v', v; v=0, \dots, n; \Gamma=0, 1) \quad i=1, \dots, n$   
 と生成する。

【0135】変換情報保有検証処理505により、この挑戦値501を用いて入力文列400と、変換情報保有コミットメント408と出力暗号文列417である出力文列406と応答416とを用いて検証式、

$$\prod_{\Gamma=0}^1 v^{-\Gamma} g[\mu, \Gamma]^{(v^{-\Gamma})} = \prod_{\Gamma=0}^1 g'[\mu, \Gamma]^{(v^{-\Gamma})} / F_p \quad \Gamma=0, 1$$

が成り立つことを確認502する。

【0136】変換条件検証処理506により、挑戦値501と応答416と変換条件コミットメント412とを用いて検証式  $v^{r[0]} v^{-\phi[0]} \{ \sum_{i=1}^n r[i] r[i] \} = \omega v^{-\phi[0]} \{ c[i] c[i] + \phi[i] c[i] \} / F_p$  が成り立つことを確認503する。

【0137】以上全ての検証式が成り立てば、証明文を受理504する。

【0138】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも正



規直交行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。

【0139】入力暗号文と出力暗号文に制限が課せられており、この効果で、再暗号シャッフルの正当性を保証できる場合には、本実施例により、証明付再暗号シャッフルを構成できる。

【0140】例えば、入力暗号文は限られた候補から選ばれていることが証明されていて、かつそれらの候補は互いに他を基底として表現できないとする。この入力暗号文を再暗号シャッフルしたのち、復号していずれの復号文も正しい候補から選ばれたものであったとき、本実施例による証明文からこの再暗号シャッフルが正当であることが言える。

【0141】なお図4における変換情報保有コミットメント処理419、変換条件コミットメント生成処理420、応答生成処理421は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図5における変換情報保有検証処理505、変換条件検証処理506は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0142】〔実施例(2)〕本発明の実施例(2)の証明付再暗号シャッフル方法およびその検証方法について、図6、図7を参照して説明する。以下では、 $\Gamma=0,1$ を取るものとする。

【0143】再暗号シャッフル情報601として、再暗号シャッフル行列602、元係数603、係数基底604、605、準元係数606を、以下のように準備する。

【0144】まず再暗号シャッフル行列602に関しては、前記実施例(1)と同様に生成する。

【0145】元係数603  $\rho', \rho''$ 、係数基底604  $v$ 、係数基底605  $u$ 、準元係数606  $\lambda[\mu]$ ;  $\mu=0, \dots, n$  に関しても、実施例(1)と同様な手法で、 $\rho', \rho'' \lambda[\mu]$ ;  $\mu=0, \dots, n$  には1,0でない $F_q$ 上の数を、係数基底 $u, v$ には位数 $q$ の $F_p$ の元を生成する。

【0146】 $\rho' \in_r Z_q, \neq 0, 1$

$\rho'' \in_r Z_q, \neq 0, 1$

$v \in_r Z_p, \neq 0, 1, \text{ s.t. } v^q = 1 / F_p$

$\lambda[\mu] \in_r Z_q, \neq 0, 1, \mu=0, \dots, n$

$u \in_r Z_p, \neq 0, 1, \text{ s.t. } u^q = 1 / F_p$

【0147】実施例(1)と同様にして、入力暗号文列と公開鍵より、入力文列600  $g[\mu, \Gamma]$ ;  $\mu=0, \dots, n$ ;  $\Gamma=0, 1$  を生成する。

【0148】以下、証明付再暗号シャッフル方法を用いる。

【0149】実施例(1)と同様に、変換情報保有コミットメント生成処理623を行い、出力文列603  $g'[\mu, \Gamma]$ ;  $\mu=0, \dots, n$ ;  $\Gamma=0, 1$  を、生成する。ここで、 $g'[\mu, \Gamma]$ ;  $\mu=0, \dots, n$ ;  $\Gamma=0, 1$  を、出力暗号文列604、 $g'[0, \Gamma]$ ;  $\Gamma=0, 1$  を変換情報保有コミットメント605とする。

【0150】変換条件コミットメント生成処理625における恒等式係数計算606により、元係数603  $\rho', \rho''$ 、と、再暗号シャッフル行列602を用いて恒等式係数607  $\psi[i], \phi[i], \phi[0], \rho', \rho''$ ;  $i=1, \dots, n$  を、  
 $\rho' = \rho'$   
 $\rho'' = \rho''$   
 $\psi[i] = \sum_{j=1}^n (3A[j, 0] + \rho'' \lambda[j]) A[j, i] / F_q, i=1, \dots, n$   
 $\phi[i] = \sum_{j=1}^n (3A[j, 0] A[j, 0] A[j, i] + 2\rho'' \lambda[j] A[j, 0] A[j, i]) + \rho' A[0, i] / F_q, i=1, \dots, n$   
 $\phi[0] = \sum_{j=1}^n (A[j, 0] A[j, 0] A[j, 0] + \rho'' \lambda[j] A[j, 0] A[j, 0]) + \rho' \lambda[0] + \rho' A[0, 0] / F_q$   
と生成する。

【0151】さらに、係数基底604  $v$  を用いて、隠蔽処理608により、恒等式係数607  $\rho', \rho'', \phi[0]$  を、

$\omega = v^{(\psi[0])} / F_p$

$v' = v^{(\rho'')} / F_p$

$v'' = v^{(\rho')} / F_p$

とコミット609する。さらに、係数基底605  $u$  を用いて、準元係数606  $\lambda[\mu]$ ;  $\mu=0, \dots, n$  を、

$u[0] = u^{(\lambda[0])} / F_q$

$u[i] = u^{(\lambda[i])} / F_q, i=1, \dots, n$

とコミット612する。

【0152】以上より、 $\psi[i], \phi[i], \omega, v', v'', v, u, u[0], u[i]$ ;  $i=1, \dots, n$  を変換条件コミットメント613とする。

【0153】ここで、コミットメント614を、変換情報保有コミットメント605と、変換条件コミットメント613とする。

【0154】応答生成処理624により、以上の入力文列600と、出力暗号文列604と、コミットメント614と、を挑戦値生成関数615の引数として、挑戦値616を、

$c[0]=1$

$c[i] = \text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], u, u[v], v, \phi[j],$

$\psi[j], \omega, v', v''; \Gamma=0, 1, 2; v=0, \dots, n; j=1, \dots, n) i=1, \dots, n$

と生成し、この挑戦値616から、再暗号シャッフル行列602を用いて、応答618を、

$r[\mu] = \sum_{v=0}^n A[\mu, v] c[v] / F_q, \mu=0, \dots, n$

と生成617する。

【0155】さらに、準応答620を、準元係数606  $\lambda[\mu]$ ;  $\mu=0, \dots, n$  と、応答618より、

$r' = \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] / F_q$

と生成619する。

【0156】以上のコミットメント614と、応答618と、



準応答620と、を再暗号シャッフル証明文622として出力し、再暗号シャッフルの結果として、出力暗号文列604を出力する。検証方法について、図6及び図7を参照して、以下に説明する。

【0157】再暗号シャッフル検証方法により、入力文列600  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ と、出力暗号文列604  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ と、再暗号シャッフル証明文622中のコミットメント614である変換情報保有コミットメント605  $g''[0, \Gamma]; \Gamma=0, 1$ と、変換条件コミットメント609、912  $\psi[i], \phi[i], \omega, v', v', v, u, u[0], u[i]; i=1, \dots, n$ と、を挑戦値生成関数704に代入して挑戦値705を、

$c[0]=1$

$c[i]=\text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], u, u[v], v, \phi[j], \psi[j], \omega, v', v'; \Gamma=0, 1, 2; v=0, \dots, n; j=1, \dots, n) i=1, \dots, n$   
と生成する。

【0158】変換情報保有検証処理710により、この挑戦値705を用いて、入力文列600と、変換情報保有コミットメント605と、出力暗号文列604である出力文列603と、応答618と、を用いて検証式、

$\Pi_{\mu=0}^n g[\mu, \Gamma]^{c[\mu, \Gamma]} = \Pi_{\mu=0}^n g''[\mu, \Gamma]^{c[\mu, \Gamma]} / F_p, \Gamma=0, 1$

が成り立つことを確認706する。

【0159】変換条件検証処理711により、挑戦値705と、応答618と、変換条件コミットメント609、612と、を用いて検証式、

$v', r', v^{(0)} v \{ \sum_{i=1}^n r[i] r[i] r[i] \} = \omega v \{ \sum_{i=1}^n (c[i] c[i] c[i] + \psi[i] c[i] c[i] + \phi[i] c[i]) \} / F_p$

と検証式707

$u' = \prod_{i=1}^n u[i]^{r[i] r[i]} / F_p$

が成り立つこと確認708する。

【0160】以上全ての検証式が成り立てば証明文を受理709する。

【0161】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。この時、出力暗号文列  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ は、

$g'[j, 0] = g[i, 0]^{v[i]} g[0, 0]^{A[i, j]} / F_p$

$g'[j, 1] = g[i, 1]^{v[i]} g[0, 1]^{A[i, j]} / F_p$

を出力した可能性を排除できない。ここで、 $w[i]$ が全て1の時が、再暗号シャッフルである。なお、 $w[i]; i=1, \dots, n$ は、 $F_q$ 上の1の三乗根のいずれかをとる。

【0162】そこで、復号文として、 $F_q$ 上の1の三乗根乗の自由度を許すか、平文に決められた記号を記して三乗根乗の自由度を消せば、本実施例をもって、証明付再暗号シャッフルを構成できる。

【0163】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理623、変換条件コミットメン

ト生成処理625、応答生成処理624は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理710、変換条件検証処理711は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピーディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0164】【実施例(3)】本発明の実施例(3)として証明付再暗号シャッフル方法およびその検証方法について、図8、図9を参照して以下に説明する。以下で、 $\Gamma=0, 1$ を取るものとする。また公開鍵は、 $\eta[-1, \Gamma], \eta[0, \Gamma]; \Gamma=0, 1$ の二組あり、どちらも同じ秘密鍵を持つとする。

【0165】再暗号シャッフル情報801として、再暗号シャッフル行列802、元係数803、805、係数基底804、806、準元係数807を以下のように準備する。

【0166】本実施例で用いる再暗号シャッフル行列802は、実施例(1)と実施例(2)のものとは大きさが異なり、 $n+2$ 行 $n+1$ 列の行列である。

【0167】この再暗号シャッフル行列802を構成するシャッフル行列は、 $A[i, j]; i, j=1, \dots, n$ であり、再暗号秘密乱数は、 $A[-1, j], A[0, j]; j=1, \dots, n$ の $2 \times n$ 成分であり、知識の隠蔽因子は、 $A[\mu, 0]; \mu=-1, \dots, n$ の $n+2$ 成分である。これらの成分を、実施例(1)と同様に生成する。

【0168】元係数803  $r'[-1], r'[0]$ 、元係数805  $\rho, \rho', \rho''$ 、係数基底804  $v$ 、係数基底806  $u$ 、準元係数807  $\lambda[\mu]; \mu=0, \dots, n$ に関しても、実施例(1)と同様な手法で、 $r'[-1], r'[0], \rho, \rho', \rho'', \lambda[\mu]; \mu=0, \dots, n$ には、1, 0でない $F_q$ 上の数を、係数基底 $u, v$ には、位数 $q$ の $F_p$ の元を生成する。

【0169】 $r'[-1] \in {}_r Z_q, \neq 0, 1$

$r'[0] \in {}_r Z_q, \neq 0, 1$

$\rho \in {}_r Z_q, \neq 0, 1$

$\rho' \in {}_r Z_q, \neq 0, 1$

$\rho'' \in {}_r Z_q, \neq 0, 1$

$v \in {}_r Z_q, \neq 0, 1, \text{ s.t. } v^q = 1 / F_p$

$\lambda[\mu] \in {}_r Z_q, \neq 0, 1 \mu=0, \dots, n$

$u \in {}_r Z_q, \neq 0, 1, \text{ s.t. } u^q = 1 / F_p$

【0170】入力暗号文列  $\eta[i, 0], \eta[i, 1]; i=1, \dots, n$ と、公開鍵  $\eta[-1, \Gamma], \eta[0, \Gamma]; \Gamma=0, 1$ より、入力文列800  $g[\mu, \Gamma]; \mu=-1, \dots, n; \Gamma=0, 1$ を、

$g[-1, \Gamma] = \eta[-1, \Gamma] \quad \Gamma=0, 1$

$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$

$g[i, \Gamma] = \eta[i, \Gamma] / F_p \quad i=1, \dots, n, \Gamma=0, 1$

とする。

10

20

30

40

50

【0171】以下証明付再暗号シャッフル方法を用いる。

【0172】変換情報保有コミットメント生成処理832における再暗号シャッフル行列作用808により、上記再暗号シャッフル行列802を入力文列800に以下の様に作用させて、出力文列809 $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ を、

$$g'[\mu, \Gamma] = \prod_{i=1}^n g[v, \Gamma]^{A[i, \mu, \Gamma]} / F_p \quad \mu=0, \dots, n, \Gamma=0, 1$$

と生成する。ここで、 $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ を、出力暗号文列810、 $g'[0, \Gamma]; \Gamma=0, 1$ を、変換情報保有コミットメント811とする。

【0173】変換条件コミットメント生成処理833、834における恒等式係数計算812、816により、元係数803、805  $r'[-1], r'[0], \rho, \rho', \rho''$ と、再暗号シャッフル行列802を用いて、恒等式係数817  $\psi[i], \phi[i], \phi[0], \rho, \rho', \rho''; i=1, \dots, n$ と、恒等式係数813  $\phi[v], r'[0], r'[-1]; v=0, \dots, n$ を計算818、812する。

【0174】 $\rho=\rho$

$$\rho'=\rho'$$

$$\rho''=\rho''$$

$$\psi[i] = \sum_{j=1}^n (3A[j, 0] + \rho'' \lambda[j]) A[j, i] / F_q \quad i=1, \dots, n$$

$$\phi[i] = \sum_{j=1}^n (3A[j, 0] A[j, 0] A[j, i] + 2\rho'' \lambda[j] A[j, 0] A[j, i]) + \rho' A[0, i] + \rho A[-1, i] / F_q \quad i=1, \dots, n$$

$$\phi[0] = \sum_{j=1}^n (A[j, 0] A[j, 0] A[j, 0] + \rho'' \lambda[j] A[j, 0] A[j, 0]) + \rho' A[0, 0] + \rho A[-1, 0] / F_q$$

$$r'[-1] = r'[-1]$$

$$r'[0] = r'[0]$$

$$\phi[0] = \sum_{j=1}^n A[j, 0] A[j, 0] + r'[0] A[0, 0] + r'[-1] A[-1, 0] / F_q$$

$$\phi[i] = 2 \sum_{j=1}^n A[j, 0] A[j, i] + r'[0] A[0, i] + r'[-1] A[-1, i] / F_q \quad i=1, \dots, n$$

【0175】さらに係数基底804  $v$ を用いて、隠蔽処理814、818により、恒等式係数813、817  $r'[-1], r'[0], \phi[0], \phi[0], \rho, \rho', \rho''$ を、

$$\omega = v^{(0)} / F_p$$

$$v' = v^{(0)} / F_p$$

$$v' = v^{(0)} / F_p$$

$$\omega' = v^{(0)} / F_p$$

とコミット819し、

$$v = v^{(1)} / F_p$$

$$v' = v^{(1)} / F_p$$

$$\omega = v^{(0)} / F_p$$

とコミット815する。

【0176】さらに、係数基底806  $u$ を用いて、準元係数807  $\lambda[\mu]; \mu=0, \dots, n$ を、

$$u[0] = u^{(0)} / F_q$$

$$u[i] = u^{(i)} / F_q \quad i=1, \dots, n$$

とコミット821、820する。

【0177】以上より、 $\phi[i], v', v, \omega, \psi[i], \phi[i], \omega, v', v', \omega', v, u, u[0], u[i]; i=1, \dots, n$ を変換条件コミットメント822とする。

【0178】ここで、コミットメント823を、変換情報保有コミットメント811と変換条件コミットメント822とする。

【0179】応答生成処理835により、以上の入力文列800と、出力暗号文列810と、コミットメント823を、挑戦値生成関数824の引数として、挑戦値825を、

$$c[0] = 1$$

$$c[i] = \text{Hash}[i](g[\mu, \Gamma], g'[v, \Gamma], u[v], u, \phi[j], \psi[j], \omega, \omega', v', v', v, \phi[j], \omega, v', v; \mu=-1, \dots, n; v=0, \dots, n; j=1, \dots, n; \Gamma=0, 1, 2) i=1, \dots, n$$

と生成し、この挑戦値825から、再暗号シャッフル行列802を用いて、応答827を、

$$r[\mu] = \sum_{v=0}^n A[\mu, v]^{c[v]} / F_q \quad \mu=-1, \dots, n$$

と生成826する。

【0180】さらに、準応答829を、準元係数807  $\lambda$

$$[\mu]; \mu=0, \dots, n \text{と応答827より、}$$

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] / F_q$$

と生成828する。

【0181】以上のコミットメント823と、応答827と、準応答829とを再暗号シャッフル証明文831として出力し、再暗号シャッフルの結果として、出力暗号文列810を出力する。

【0182】検証方法について図9を参照して説明する。

【0183】再暗号シャッフル検証方法により、入力文列800  $g[\mu, \Gamma]; \mu=-1, \dots, n; \Gamma=0, 1$ と、出力暗号文列810  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ と、再暗号シャッフル証明文831中コミットメント823の変換情報保有コミットメント811  $g'[0, \Gamma]; \Gamma=0, 1$ と、変換条件コミットメント815、819、821  $\phi[i], v', v, \omega, \psi[i], \phi[i], \omega, v', v', \omega', v, u, u[0], u[i]; i=1, \dots, n$ とを、挑戦値生成関数900に代入して、挑戦値901を、

$$c[0] = 1$$

$$c[i] = \text{Hash}[i](g[\mu, \Gamma], g'[v, \Gamma], u[v], u, \phi[j], \psi[j], \omega, \omega', v', v', v, \phi[j], \omega, v', v; \mu=-1, \dots, n; v=0, \dots, n; j=1, \dots, n; \Gamma=0, 1, 2) i=1, \dots, n$$

と生成する。

【0184】変換情報保有検証処理907により、この挑戦値901を用いて入力文列800と、変換情報保有コミットメント811と、出力暗号文列810である出力文列809と、応答827とを用いて検証式、

$$\prod_{\mu=-1}^n g[\mu, \Gamma]^{r[\mu]} = \prod_{\mu=0}^n g'[\mu, \Gamma]^{c[\mu]} / F_p \quad \Gamma=0, 1$$

が成り立つことを確認902する。

【0185】変換条件検証処理908、909により、挑戦値901と応答827と準応答829と変換条件コミットメント81

50 5、819、821とを用いて、検証式

$$v^{r',r'} = v^{r(0)} \omega^{r(-1)} v^{\{\sum_{i=1}^n r[i]r[i]r[i]\}} \\ = \omega v^{\{\sum_{i=1}^n (c[i]c[i]c[i] + \phi[i]c[i]c[i] + \phi[i]c[i]c[i])\}} / F_p$$

が成り立つことを確認904し、検証式

$$u^{r'} = u[0] \prod_{i=1}^n u[i]^{r(i)r(i)} / F_p$$

が成り立つことを確認905し、検証式

$$\underline{v}^{r(0)} \underline{v}^{r(-1)} v^{\{\sum_{i=1}^n r[i]r[i]\}} = \omega v^{\{\sum_{i=1}^n (c[i]c[i] + \phi[i]c[i])\}} / F_p$$

が成り立つことを確認903する。

【0186】以上全ての検証式が成り立てば証明文を受 10 理906する。

【0187】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【0188】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理832、変換条件コミットメント生成処理833、834、応答生成処理835は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理907、変換条件検証処理908、909は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0189】【実施例(4)】本発明の実施例(4)の証明付再暗号シャッフル方法およびその検証方法について図10、図11を参照して以下に説明する。以下で、 $\Gamma=0,1$ を取るものとする。また公開鍵は、 $\eta[0, \Gamma]; \Gamma=0,1$ の一组である。

【0190】再暗号シャッフル情報1006として、再暗号シャッフル行列1001と、第2の情報隠蔽因子1004と、元係数1002、1005と、係数基底1003、1008と、準元係数1007を以下のように準備する。

【0191】まず再暗号シャッフル行列1001に関しては、前記実施例(1)と同様に生成し、これを $A[\mu, v]; \mu, v=0, \dots, n$ とする。

【0192】さらに、第2の情報隠蔽因子1004  $A[v, 0]; v=0, \dots, n$ を同様に生成する。

【0193】元係数1005  $\rho', \rho''$ 、元係数1002  $r'$   $[0]$ 、係数基底1003  $v$ 、係数基底1008  $u$ 、準元係数1007

$\lambda[\mu]; \mu=0, \dots, n$  に関しても、実施例(1)と同様な手法で、 $r'[0], \rho', \rho'', \lambda[\mu]; \mu=0, \dots, n$ には1,0でない $F_q$ 上の数を、係数基底 $u, v$ には位数 $q$ の $F_p$ の元を生成 50

する。

【0194】 $\rho' \in {}_R Z_q, \neq 0, 1$

$\rho'' \in {}_R Z_q, \neq 0, 1$

$r'[0] \in {}_R Z_q, \neq 0, 1$

$v \in {}_R Z_p, \neq 0, 1, \text{ s.t. } v^q = 1 / F_p$

$\lambda[\mu] \in {}_R Z_q, \neq 0, 1 \mu=0, \dots, n$

$u \in {}_R Z_p, \neq 0, 1, \text{ s.t. } u^q = 1 / F_p$

【0195】入力暗号文列  $\eta[i, 0], \eta[i, 1]; i=1, \dots, n$ と、公開鍵  $\eta[0, \Gamma]; \Gamma=0,1$ より、入力文列1000  $g[\mu \Gamma]; \mu=0, \dots, n; \Gamma=0,1$ を、

$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0,1$

$g[i, \Gamma] = \eta[i, \Gamma] \quad i=1, \dots, n, \Gamma=0,1$

とする。

【0196】以下、証明付再暗号シャッフル方法を用いる。

【0197】変換情報保有コミットメント生成処理1042における再暗号シャッフル行列作用1009により、上記暗号シャッフル行列1001を入力文列1000に以下の様に作用させて、出力文列1010  $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0,1$ を、

$$g'[\mu, \Gamma] = \prod_{v=0}^n g[v, \Gamma]^{A[\mu, v]} / F_p \quad \mu=0, \dots, n, \Gamma=0,1$$

と生成する。

【0198】ここで、 $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0,1$ 、出力暗号文列1011、 $g'[0, \Gamma]; \Gamma=0,1$ を第1の変換情報保有コミットメント1012とする。

【0199】第2の変換情報保有コミットメント生成処理1044により、さらに入力文列1000から選択1018して、第2の入力文列1019を $g[\mu, \Gamma']$ とする。ここでは、 $\Gamma'=0$ とする。

【0200】第2の変換情報保有コミットメント1021  $g''[0, \Gamma']$ を、 $g''[0, \Gamma'] = \prod_{v=0}^n g[v, \Gamma']^{A[0, v]} / F_p \quad \Gamma'=0 \text{ or } 1$ と生成1020する。

【0201】変換条件コミットメント生成処理1045における恒等式係数計算1022により、元係数1005  $\rho', \rho''$ と、再暗号シャッフル行列1001と、を用いて、恒等式係数1023  $\phi[i], \phi[i], \phi[0], \rho', \rho''; i=1, \dots, n$ を、 $\rho' = \rho'$

$$\rho'' = \rho''$$

$$\phi[i] = \sum_{j=1}^n (3A[j, 0] + \rho'' \lambda[j]) A[j, i] / F_q \quad i=1, \dots, n$$

$$\phi[i] = \sum_{j=1}^n (3A[j, 0] A[j, 0] A[j, i] + 2\rho'' \lambda[j] A[j, 0] A[j, i]) + \rho' A[0, i] / F_q \quad i=1, \dots, n$$

$$\phi[0] = \sum_{j=1}^n (A[j, 0] A[j, 0] A[j, 0] + \rho'' \lambda[j] A[j, 0] A[j, 0]) + \rho' \lambda[0] + \rho' A[0, 0] / F_q$$

と生成する。

【0202】さらに、係数基底1003  $v$ を用いて、隠蔽処理1024により、恒等式係数1023  $\phi[0], \rho', \rho''$ を、 $\omega = v^{q(0)} / F_p$

$$v' = v^{(0)} / F_p$$

$$v' = v^{(0)} / F_p$$

とコミット1025する。

【0203】さらに、係数基底1008uを用いて、準元係数1007 $\lambda[\mu]$ ;  $\mu=0, \dots, n$ を、

$$u[0] = u^{(0)} / F_q$$

$$u[i] = u^{(i)} / F_q \quad i=1, \dots, n$$

とコミット1027する。

【0204】変換条件コミットメント生成処理1043における恒等式係数計算1013により、元係数1002 $r'[0]$ と再暗号シャッフル行列1001と第2の情報隠蔽因子1004とを用いて恒等式係数1014  $\phi[v], r'[0]$ ;  $v=0, \dots, n$ を、 $r'[0] = r'[0]$

$$\phi[0] = \sum_{j=1}^n A[j, 0] A[j, 0] + r'[0] A[0, 0] / F_q$$

$$\phi[i] = 2 \sum_{j=1}^n A[j, 0] A[j, i] + r'[0] A[0, i] / F_q \quad i=1, \dots, n$$

と生成する。

【0205】さらに係数基底1003  $v$ を用いて、隠蔽処理1015により、恒等式係数1014  $r'[0], \phi[0]$ を、

$$v' = v^{(0)} / F_p$$

$$\omega = v^{(0)} / F_p$$

とコミット1016する。

【0206】以上により、第1の変換条件コミットメント1028を、 $\phi[i], \phi[i], \omega, v', v', v, u, u[0], u[i]$ ;  $i=1, \dots, n$ とする。第2の変換条件コミットメント1016を、 $\phi[i], v', \omega, v$ ;  $i=1, \dots, n$ とする。

【0207】ここで、第1のコミットメント1017を、第1の変換情報保有コミットメント1012と第1の変換条件コミットメント1028とし、第2のコミットメント1029を第2の変換情報保有コミットメント1021と第2の変換条件コミットメント1016とする。

【0208】応答生成処理1046により、以上の入力文列1000と、出力暗号文列1011と、第1コミットメント1017とを、挑戦値生成関数1030の引数として、第1の挑戦値1031を、

$$c[0] = 1$$

$$c[i] = \text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], u[v], u, \phi[j], \phi[j], \omega, v', v', v; v=0, \dots, n; j=1, \dots, n; \Gamma=0, 1) \quad i=1, \dots, n$$

と生成し、この挑戦値1031から、再暗号シャッフル行列1001を用いて、第1の応答1033を、

$$r[\mu] = \sum_{v=0}^n A[\mu, v] c[v] / F_q \quad \mu=0, \dots, n$$

と生成1032する。

【0209】さらに、準応答1039を、準元係数1007 $\lambda$

$$[\mu]; \mu=0, \dots, n$$
と、応答1033より、

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] / F_q$$

と生成1038する。

【0210】応答生成処理1047により、第2の入力文列1019と、出力暗号文列1011と、第2コミットメント1029とを、挑戦値生成関数1034の引数として、第2の挑戦値

1035を、

$$c[0] = 1$$

$$c[i] = \text{Hash}[i](g[v, \Gamma], g'[0, \Gamma], g'[j, \Gamma], \phi[j], \omega, v'; v=0, \dots, n; j=1, \dots, n; \Gamma'=0) \quad i=1, \dots, n$$

と生成し、この挑戦値1035から、再暗号シャッフル行列1001と、第2の情報隠蔽因子1004を用いて、第2の応答1037を、

$$r[\mu] = A[\mu, 0] + \sum_{i=1}^n A[\mu, i] c[i] / F_q \quad \mu=0, \dots, n$$

と生成1036する。

【0211】以上のコミットメント1017、1029と、応答1033、1037と、準応答1039と、を、再暗号シャッフル証明文1040として出力し、再暗号シャッフルの結果として出力暗号文列1011を出力する。

【0212】検証方法について、図11を参照して説明する。

【0213】再暗号シャッフル検証方法により、入力文列1000と、出力暗号文列1011と、再暗号シャッフル証明文1040の第1のコミットメント1012、1025、1027を、挑戦値生成関数1100に代入して、第1の挑戦値1101を、

$$c[0] = 1$$

$$c[i] = \text{Hash}[i](\text{入力文列、出力暗号文列、第1のコミットメント}) \quad i=1, \dots, n$$
と生成する。

【0214】さらに、第2の入力文列1019と、再暗号シャッフル証明文1040の第2のコミットメント1016、1021と、出力暗号文列1011とを、挑戦値生成関数1108に代入して第2の挑戦値1109を

$$c[0] = 1$$

$$c[i] = \text{Hash}[i](\text{第2の入力文列、出力暗号文列、第2のコミットメント}) \quad i=1, \dots, n$$
と生成する。

【0215】変換情報保有検証処理1112により、第1の挑戦値1101を用いて、入力文列1000と、第1の変換情報保有コミットメント1012と、出力暗号文列1011と、第1の応答1033と、を用いて検証式、

$$\Pi_{\mu} \circ g[\mu, \Gamma]^{r[\mu]} = \Pi_{\mu} \circ g'[\mu, \Gamma]^{c[\mu]} / F_p, \quad \Gamma=0, 1$$

が成り立つことを確認1103する。

【0216】変換情報保有検証処理1113により、第2の挑戦値1109を用いて第2の入力文列1019と、第2の変換情報保有コミットメント1021と、出力暗号文列1011と、第2の応答1037と、を用いて第2の知識検証式、

$$\Pi_{\mu} \circ g[\mu, \Gamma']^{r[\mu]} = g'[0, \Gamma'] \Pi_{i=1}^n g'[i, \Gamma']^{c[i]} / F_p, \quad \Gamma'=0$$

が成り立つことを確認1105する。

【0217】変換条件検証処理1111により、第1の挑戦値1101と、第1の応答1033と、第1の変換条件コミットメント1025とを用いて、検証式1102、

$$v^{r'} \cdot v^{r[0]} \cdot \omega^{r[1]} \cdot v^{-\{\sum_{i=1}^n r[i] r[i] r[i]\}} = \omega v^{-\{\sum_{i=1}^n (c[i] c[i] c[i] + \phi[i] c[i] c[i] + \phi[i] c[i])\}} / F_p$$

と、準応答1039と、準応答コミットメント1027と、第1

の応答1033と、検証式1107、  

$$u' = u[0] \prod_{i=1}^n u[i]^{r(i)} / F_p$$

が成り立つことを確認する。

【0218】変換条件検証処理1114により、第2の挑戦値1109と、第2の応答1037と、第2の変換条件コミットメント1016とを用いて、検証式1106、

$$v' = v \{ \sum_{i=1}^n r[i] c[i] \} = \omega v \{ \sum_{i=1}^n (c[i] c[i] + \phi [i] c[i]) \} / F_p$$

が成り立つことを確認する。

【0219】以上全ての検証式が成り立てば証明文を受  
理1110する。

【0220】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【0221】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理1042、変換条件コミットメント生成処理1043、1045、応答生成処理1046、1047は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理1112、1113、変換条件検証処理1111、1114は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0222】〔実施例(5)〕本発明の実施例(5)をなす入力文列生成方法について、図12を参照して、以下に説明する。 $\Gamma$ は、0、1、2の値をとる。

【0223】公開鍵302  $g[0,0]$ 、 $g[0,1]$ に対応する秘密鍵 $x$ は、 $t$ 人の証明者により分散所持されている。

【0224】この時、秘密鍵 $x[\Lambda]$ ； $\Lambda=1, \dots, t$ により、各証明者の公開鍵を、 $g[0,0]$ 、 $g[0,1, \Lambda]=g[0,0]^{x[\Lambda]}$ ； $\Lambda=1, \dots, t$ とし、全体の公開鍵は、 $g[0,0]$ 、 $g[0,1]=\prod_{i=1}^t g[0,1, \Lambda]$ とする。

【0225】入力暗号文列301  $\eta[i,0]$ 、 $\eta[i,1]$ ； $i=1, \dots, n$ と、公開鍵302  $\eta[0,0]$ 、 $\eta[0,1]$ が入力され、公開鍵302およびElgamal領域変数 $p$ 、 $q$ より、基底生成関数1200で、入力ベクトル1201を生成し、入力文列300  $g[\mu, \Gamma]$ ； $\mu=0, \dots, n$ ； $\Gamma=0, 1, 2$ を、

$$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$$

$$g[i, \Gamma] = \eta[i, \Gamma] / F_p \quad i=1, \dots, n, \quad \Gamma=0, 1$$

$$g[\mu, 2] = \text{Hash}'[\mu](p, q, \eta[0,0], g[0,1, \Lambda]; \Lambda=1, \dots, t) \mu=0, \dots, n$$

とする。

【0226】前記実施例(1)乃至実施例(4)に、本実施例の入力文列生成方法を適用する場合、これらの実施例における $\Gamma$ の値をとる範囲を、全て0、1から、0、1、2に変更する。この新たに導入された、入力暗号文でも公開鍵でもない $\Gamma=2$ の成分が、入力暗号文生成者にも意図できない入力文列の成分となり、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【0227】また前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力文列を、 $g[-1, \Gamma]$ まで拡張して、公開鍵  $g[-1,0]$ 、 $g[-1,1]$ 、 $\eta[0,0]$ 、 $\eta[0,1]$ より、

$$g[-1, \Gamma] = \eta[-1, \Gamma] \quad \Gamma=0, 1$$

$$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$$

$$g[i, \Gamma] = \eta[i, \Gamma] / F_p \quad i=1, \dots, n, \Gamma=0, 1$$

$$g[\mu, 2] = \text{Hash}'[\mu](p, q, \eta[0,0], g[0,1, \Lambda]; \Lambda=1, \dots, t) \mu=-1, \dots, n$$

とする。

【0228】また実施例(4)に、本実施例の入力文列生成方法を適用する場合、 $\Gamma'=2$ とし、第2の情報隠蔽因子より、第2の変換情報保有コミットメントを、

$$g'[0,2] = \prod_{i=1}^n g[v, 2]^{A^{(v, 0)}} / F_p$$

$$g'[i,2] = \prod_{i=1}^n g[v, 2]^{A^{(v, i)}} / F_p \quad i=1, \dots, n$$

と変更する。

【0229】さらに、証明付再暗号シャッフル方法または再暗号シャッフル検証方法において、第2の入力文列  $g[\mu, \Gamma']$ ； $\Gamma=2$ と、第2のコミットメントとを挑戦値生成関数の引数として、第2の挑戦値を、

$$c[0]=1$$

$$c[i] = \text{Hash}[i](g[v, 2], g'[v, 2], \phi[j], \omega, v'; v=0, \dots, n; j=1, \dots, n; \Gamma=0, 1) \quad i=1, \dots, n$$

と変更する。

【0230】さらに、変換情報保有検証処理における第2の知識検証式は、

$$\prod_{i=1}^n g[\mu, 2]^{r[\mu, i]} = \prod_{i=1}^n g'[\mu, \Gamma']^{c[\mu, i]} / F_p$$

と変更する。

【0231】〔実施例(6)〕本発明の実施例(6)をなす入力文列生成方法について、図13と図14を参照して説明する。 $\Gamma$ は、0、1の値をとる。

【0232】前記実施例(5)と同様に、秘密鍵 $x$ は、 $t$ 人の証明者により分散所持されている。各証明者 $\Lambda$ ； $\Lambda=1, \dots, t$ は、証明付公開鍵列方法1304により、入力暗号文列301  $\eta[i,0]$ 、 $\eta[i,1]$ ； $i=1, \dots, n$ と、公開鍵302  $\eta[0,0]$ 、 $\eta[0,1]$ とを、共通初期値1310として、秘密鍵1301 $x[\Lambda]$ と、疑似秘密鍵1302 $\alpha[\Lambda]$ と、を公開鍵列情報1300として入力し、分散公開鍵列対1305  $g'[\mu, 1, \Lambda]$ ； $\mu=0, \dots, n$ と、公開鍵列証明文1306と、を得る。

【0233】公開鍵列検証方法1307により、各証明者の出力した分散公開鍵列対1305と、公開鍵列証明文と、共

通初期値1310とから、分散公開鍵列1305の正当性が検証されたら、前処理方法により、各証明者の分散公開鍵列対1305  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n; \Lambda=1, \dots, t$ を合わせて、公開鍵列対140  $3g'[\mu, 1, \Lambda]; \mu=0, \dots, n$ を、 $g'[\mu, 1]=\Pi_{\Lambda=1}^t g'[\mu, 1, \Lambda] / F_p$   $\mu=0, \dots, n$ とする。ここで、 $g'[0, 1]=\eta[0, 1]$ に入れ替える。

【0234】共通初期値である入力暗号文列301  $\eta[i, 0]$ 、 $\eta[i, 1]; i=1, \dots, n$ と、公開鍵302  $\eta[0, 0]$ 、 $\eta[0, 1]$ とから、公開鍵列底1401  $g'[\mu, 0]; \mu=0, \dots, n$ を、 $g'[0, 0]=\eta[0, 0]$   
 $g'[i, 0]=\text{Hash}'[i](\eta[0, 0], \eta[0, 1, \Lambda], \eta[j, \Gamma]; \Lambda=1, \dots, t; \Gamma=0, 1; j=1, \dots, n; i=1, \dots, n)$   
 と生成1400する。ここでも、公開鍵列対1403と同様に $g'[0, 0]$ が入れ替えている。

【0235】公開鍵列底1401と公開鍵列対1403を合わせて、公開鍵列1404  $g'[\mu, \Gamma]; \mu=0, \dots, n, \Gamma=0, 1$ とする。

【0236】公開鍵列1404と、入力暗号文列301と、公開鍵302とから、入力文列300  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ を、  
 $g[0, \Gamma]=\eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma]=\eta[i, \Gamma]g'[i, \Gamma] / F_p \quad i=1, \dots, n, \Gamma=0, 1$   
 とする(前処理1402)。

【0237】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力暗号文列  $\eta[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ と、公開鍵  $\eta[0, \Gamma]; \Gamma=0, 1$ に対して、公開鍵列  $g'[\mu, \Gamma]; \mu=1, \dots, n; \Gamma=0, 1$ を生成する。ただし、 $g'[0, \Gamma]; \Gamma=0, 1$ は、公開鍵に等しい。そして、入力文列  $g[\mu, \Gamma]; \mu=1, \dots, n; \Gamma=0, 1$ を、  
 $g[-1, \Gamma]=\eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma]=\eta[i, \Gamma]g'[i, \Gamma] / F_p \quad i=0, \dots, n, \Gamma=0, 1$   
 とする。

【0238】本実施例では、新たに生成された公開鍵列が、入力暗号文生成者にも意図できないため、入力暗号文にそれを乗じた入力文列の成分も意図できない。そのため、証明者が生成できる応答に制限を課す働きをなし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【0239】なお図13に示した、証明付公開鍵列装置1304、前処理装置1309、公開鍵列検証装置1307は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体(例えばCD-ROM、DVD(digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか)から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0240】〔実施例(7)〕本発明の実施例(7)の入力文列生成方法について、図15乃至図18を参照して説明す

る。 $\Gamma$ は、0、1の値をとる。前記実施例(5)と同様に、秘密鍵1502  $x$ は、 $t$ 人の証明者により、分散所持されている。

【0241】各証明者 $\Lambda; \Lambda=1, \dots, t$ は、証明付公開鍵列方法1504により、Elgamal領域変数を共通初期値1500として、秘密鍵1502  $x[\Lambda]$ と、疑似秘密鍵1503  $\alpha[\Lambda]$ と、を、公開鍵列情報1501として入力し、分散公開鍵列対1505  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n$ と、公開鍵列証明文1506と、を得る。

10 【0242】公開鍵列検証方法1507により、各証明者の出力した分散公開鍵列対1505と、公開鍵列証明文1506と、共通初期値1500より、分散公開鍵列対1505の正当性が検証1508されたら、各証明者の分散公開鍵列対1505  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n; \Lambda=1, \dots, t$ を合わせて、公開鍵列対1509  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n$ を、  
 $g'[\mu, 1]=\Pi_{\Lambda=1}^t g'[\mu, 1, \Lambda] / F_p \quad \mu=0, \dots, n$   
 とする。

【0243】共通初期値1500から公開鍵列底 $g'[\mu, 0]; \mu=0, \dots, n$ を、  
 20  $g'[\mu, 0]=\text{Hash}'[\mu](p, q) \quad \mu=0, \dots, n$   
 と生成する。

【0244】公開鍵列底と公開鍵列対1509を合わせて、公開鍵列1611  $g'[\mu, \Gamma]; \mu=0, \dots, n, \Gamma=0, 1$ とする。

【0245】各入力暗号文生成者 $i=1, \dots, n$ は、証明付暗号化方法1606により、平文1602m[i]と、個別公開鍵1601g[i,  $\Gamma$ ];  $\Gamma=0, 1$ と、秘密乱数1604s[i]と、疑似秘密乱数1605s'[i]とより、入力暗号文1607  $\eta[i, \Gamma]; \Gamma=0, 1$ を、

30  $\eta[i, 0]=g'[i, 0]^{s[i]} / F_p$   
 $\eta[i, 1]=m[i]g'[i, 1]^{s[i]} / F_p$   
 と生成する。

【0246】またコミットメント(疑似暗号文底1704)、挑戦値1707、応答1709を順に以下のように生成し、  
 $\eta[i2]=g'[i, 0]^{s[i]} / F_p$   
 $c'[i]=\text{Hash}[0](\eta[i, 0], \eta[i, 1], \eta[i2])$   
 $\theta'[i]=c'[i]s[i]+s'[i] / F_p$   
 疑似暗号文底1704と応答1709とを暗号化証明文1608とする。

【0247】暗号化検証装置により、全ての入力暗号文1607と暗号化証明文1608に関して、  
 40  $c'[i]=\text{Hash}[0](\eta[i, 0], \eta[i, 1], \eta[i2])$   
 と、挑戦値1801を求め、これに応答1709を用いて、検証式1802

$$\eta[i, 0]^{c'[i]} = \eta[i, 1]^{c'[i]} \eta[i2] / F_p$$

が成り立つことを確認1610する。入力暗号文1607全ての正当性が確認されたら、入力暗号文323  $\eta[i, \Gamma]; \Gamma=0, 1$ と、共有公開鍵1600  $g'[0, \Gamma]; \Gamma=0, 1$ とより入力文列300を、

$$g[0, \Gamma]=g'[0, \Gamma]$$

50  $g[i, \Gamma]=\eta[i, \Gamma] \quad i=1, \dots, n$

とする。

【0248】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合は、

$$g[-1, \Gamma] = g'[-1, \Gamma]$$

$$g[0, \Gamma] = g'[0, \Gamma]$$

$$g[i, \Gamma] = \eta[i, \Gamma] \quad i=1, \dots, n$$

とする。

【0249】本実施例では、始めに生成された公開鍵列が、入力暗号文生成者にも意図できないため、これを基に暗号化したことが主命されている入力暗号文の成分も、意図できない。そのため、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【0250】なお図15に示した、証明付公開鍵列装置1504、公開鍵列検証装置1507は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図16乃至図18に示した、証明付暗号化装置1606、暗号化検証装置1609は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0251】〔実施例(8)〕本発明の実施例(8)における証明付公開鍵列方法について図19と図20とを参照して説明する。

【0252】共通初期値 $e$ と、秘密鍵 $1902 \ x$ と、疑似秘密鍵 $1903 \ \alpha$ とが、公開鍵列情報 $1901$ として、入力される。

【0253】共通初期値 $1900$ から、公開鍵列底 $1905 \ g'[\mu, 0]; \mu=0, \dots, n$ を、 $g'[\mu, 0] = \text{Hash}'[\mu](e) \quad \mu=0, \dots, n$ と生成 $1904$ する。

【0254】これから、秘密鍵 $1902 \ x$ と、疑似秘密鍵 $1903 \ \alpha$ とにより、(分散)公開鍵列対 $1907 \ g'[\mu, 1]; \mu=0, \dots, n$ が、 $g'[\mu, 1] = g'[\mu, 0]^x / F_p \quad \mu=0, \dots, n$ と生成 $1906$ され、疑似公開鍵列対 $1909$ が、 $g'[\mu, 2] = g'[\mu, 0]^\alpha / F_p \quad \mu=0, \dots, n$ と生成 $1908$ される。

【0255】挑戦値 $1912$ 、応答 $1914$ を順に、 $c' = \text{Hash}[0](g'[\mu, 0], g'[\mu, 2]) \quad \mu=0, \dots, n$   
 $\theta = c' \cdot x + \alpha / F_q$ と生成し、疑似公開鍵列 $1909$ と応答 $1914$ を公開鍵列証明文 $1915$ とする。

【0256】公開鍵列検証方法により、挑戦値 $2003$ を、 $c' = \text{Hash}[0](g'[\mu, 0], g'[\mu, 2]); \mu=0, \dots, n$

と生成 $2000$ し、応答 $1914$ を用いて検証式、 $g'[\mu, 0]^\theta = g'[\mu, 0]^{c'} \cdot g'[\mu, 2] / F_p \quad \mu=0, \dots, n$ を検証 $2004$ する。

【0257】本実施例では、始めに生成された公開鍵列底、誰にも意図できないため、これを元に作られた公開鍵列の成分も意図できない。

【0258】なお図19、図20に示した、証明付公開鍵列装置、公開鍵列検証装置は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0259】〔実施例(9)〕本発明の実施例(9)として、証明付復号について説明する。前記実施例(5)と同様に、秘密鍵 $x$ は、 $t$ 人の証明者により分散所持されている。

【0260】 $\Lambda; \Lambda=1, \dots, t$ 番目の証明者は、 $\Lambda-1$ 番目の証明者による部分復号の結果を入力して、それを部分復号する。 $\Lambda$ 番目の証明者による部分復号の結果が復号文である。ただし、 $0$ 番目の証明者による部分復号の結果は、上記最終的な再暗号シャッフルの出力のことである。

【0261】ここで、 $\Lambda$ 番目の証明者の行う証明付部分復号(部分復号およびその証明文の提出)について説明する。

【0262】疑似乱数発生器により $1, 0$ でない $F_q$ 上の数を $\beta[\Lambda]$ を作成する。

【0263】 $\beta[\Lambda] \in_r \mathbb{Z}_q, \neq 0, 1$

【0264】また、自身の公開鍵 $g[0, 0], g'[0, 1, \Lambda]$ を、 $g[0, 0], g[0, 1]$ と、入力された暗号文列を、 $g[i, \Gamma]; i=1, \dots, n, \Gamma=0, 1$ とし、自身の公開鍵と秘密鍵 $x[\Lambda]$ から、部分復号基底 $G[\mu, 0, \Lambda]; \mu=0, \dots, n$ と、疑似部分復号基底 $G[\mu, 1, \Lambda]; \mu=0, \dots, n$ を、 $G[\mu, 0, \Lambda] = g[\mu, 0]^{x[\Lambda]} / F_p \quad \mu=0, \dots, n$   
 $G[\mu, 1, \Lambda] = g[\mu, 0]^{\beta[\Lambda]} / F_p \quad \mu=0, \dots, n$ と生成する。コミットメントとして、 $g[\mu, \Gamma, \Lambda]; \mu=0, \dots, n, \Gamma=0, 1, \Lambda=0, \dots, t$ を出力する。

【0265】 $g[0, 1, \Lambda] = g[0, 0]^{x[\Lambda]} = G[0, 0, \Lambda]$ は、公開鍵と重複しているが同じものが計算されている。

【0266】挑戦値を、 $c[\Lambda] = \text{Hash}[0](g[\mu, 0], G[\mu, \Gamma, \Lambda]); \mu=0, \dots, n; \Gamma=0, 1)$

と生成し、これを用いて、応答 $r[\Lambda]$ を

$$r[\Lambda] = \beta[\Lambda] + c[\Lambda]x[\Lambda] / F_q$$

と生成して出力する。部分復号基底、疑似部分復号基底と応答を証明付部分復号の証明文として出力する。

【0267】部分復号を



$g[i, 0] \rightarrow g[i, 0] \quad i=1, \dots, n$   
 $g[i, 1] \rightarrow g[i, 1]/G[i, 0, \Lambda] \quad /F_p \quad i=1, \dots, n$   
 として出力する。

【0268】検証処理は、入力暗号文列と証明文中とより挑戦値を、  
 $c[\Lambda] = \text{Hash}[0](g[\mu, 0], G[\mu, \Gamma, \Lambda]; \mu=0, \dots, n; \Gamma=0, 1)$

と生成し、証明文中の応答、入力暗号文列、分復号基

底、疑似部分復号基底を用いて、

$$g[\mu, 0]^{r[\Lambda]} = G[\mu, 0, \Lambda]^{c[\Lambda]} G[\mu, 1, \Lambda] \quad /F_p \quad \mu = 0, \dots, n$$

$$\begin{aligned} \prod_{\mu=1}^{n+m} g[\mu, \Gamma] r[\mu] &= \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\sum_{v=1}^{n+m} A[\mu, v] c[v]} \\ &= \prod_{v=1}^{n+m} \left( \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, v] c[v]} \right)^{c[v]} / F_p \\ &= \prod_{v=1}^{n+m} g'[\nu, \Gamma]^{c[v]} / F_p \end{aligned}$$

より分かる。

【0272】準元係数をコミットしたものと、これに付随する応答と準応答が検証式を満たすことは、

$$\begin{aligned} u' &= u^{\lambda} \{ \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] \} / F_p \\ &= u^{\lambda} \prod_{i=1}^n (u^{\lambda[i]})^{r[i] r[i]} / F_p \\ &= u[0] \prod_{i=1}^n u[i]^{r[i] r[i]} / F_p \end{aligned}$$

$$\begin{aligned} v' &= v^{\lambda} \prod_{i=1}^n v^{r[i] r[i]} / F_p \\ &= (v^{\lambda})^{\sum_{i=1}^n r[i] r[i]} v^{\sum_{i=1}^n \{ \sum_{\mu=1}^n \sum_{v=1}^n A[i, \mu] A[i, v] c[\mu] c[v] \} } / F_p \\ &= v^{\lambda} \{ r'[0] \sum_{\mu=1}^n [0, \mu] c[\mu] + 2 \sum_{i=1}^n \sum_{j=1}^n A[i, 0] A[i, j] c[j] + \sum_{i=1}^n A[i, 0] A[i, 0] + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[i, j] A[i, k] c[j] c[k] \} / F_p \\ &= v^{\lambda} \{ \sum_{i=1}^n \phi[i] c[i] + \phi[0] + \sum_{i=1}^n c[i] c[i] \} / F_p \\ &= \omega v^{\lambda} \{ \sum_{i=1}^n (c[i] c[i] + \phi[i] c[i]) \} / F_p \end{aligned}$$

より成立することがわかる。以上、 $A[i, j]$ が置換行列であるという事実を使った。

【0275】前記実施例(2)の恒等式係数に関しては、★30

$$\begin{aligned} &\sum_{i=1}^n r[i] r[i] r[i] + \sum_{i=1}^n \rho'' \lambda[i] r[i] r[i] + \rho' r[0] / F_q \\ &= \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[h, i] A[h, j] A[h, k] c[i] c[j] c[k] \\ &\quad + \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n (3A[h, 0] A[h, i] A[h, j] + \rho'' \lambda[h] A[h, i] A[h, j]) c[i] c[j] \\ &\quad + \sum_{h=1}^n \sum_{i=1}^n (3A[h, 0] A[h, 0] A[h, i] + 2\rho'' \lambda[h] A[h, 0] A[h, i] + \rho' A[0, i]) c[i] \\ &\quad + \sum_{h=1}^n (A[h, 0] A[h, 0] A[h, 0] + \rho'' \lambda[h] A[h, 0] A[h, 0] + \rho' \lambda[0] + \rho' A[0, 0]) \\ &\quad / F_q \\ &= \sum_{h=1}^n (c[h] c[h] c[h] + \psi[h] c[h] c[h] + \phi[i] c[i] + \phi[0]) / F_q \end{aligned}$$

であり、これは、 $v^{\lambda} \{ \sum_{h=1}^n (c[h] c[h] c[h] + \psi[h] c[h] c[h] + \phi[i] c[i] + \phi[0]) \} \omega[0] / F_p$ の指数部と等しい。

【0276】以上で最後の式を導くために $A[i, j]$ が置換行列であるという事実を使った。

【0277】前記実施例(3)、及び実施例(4)に関しても同様の議論より分かる。

【0278】前記実施例(8)の証明付公開鍵列方法が出力した公開鍵列底と、公開鍵列対と疑似公開鍵列対と、これに付随する応答と挑戦値が検証処理の検証式を満たすことは、

$$\begin{aligned} g'[\mu, 0] &= g'[\mu, 0]^{\alpha} / F_p \\ &= g'[\mu, 0]^{\alpha} g'[\mu, 0] / F_p \\ &= g'[\mu, 1] g'[\mu, 2] / F_p \end{aligned}$$

\*を確認し、さらに部分復号が、この $G[\mu, 0, \Lambda]$ を用いて行われたことを確認して受理する。

【0269】以上を、 $t$ 人の証明者全てにより行われた結果を復号文とする。

【0270】[正当性] 以上説明した実施例の正当性について説明する。

【0271】[完全性] 入力文列と、出力暗号文列と変換情報保有コミットメントとである出力文列と、これに付随する応答と挑戦値が変換情報保有検証処理の検証式を満たすことは、

※よりわかる。

【0273】変換条件コミットメント生成処理が出力した恒等式の係数と、これに付随する応答と挑戦値が知識検証処理の検証式を満たすことは、以下のようにして分かる。

※ 【0274】実施例(1)の恒等式係数に関しては、

$$\star v^{\lambda} v^{\lambda} \prod_{i=1}^n v^{r[i] r[i] r[i]} / F_p$$

の $v$ に対する指数部が、

から分かる。

【0279】[健全性] 与えられた挑戦値 $c[v]$ ;  $v=1, \dots, n+m'$ に対する、変換情報保有検証処理における検証式を満たす応答 $r[\mu]$ ;  $\mu=1, \dots, n+m$ を求めるには $A[\mu, v]$ ;  $\mu=1, \dots, n+m$ ;  $v=1, \dots, n+m'$ を知らなければならない。

【0280】これは、与えられた  $g[\mu, \Gamma]$ 、 $g'[\nu, \Gamma]$ ;  $\mu=1, \dots, n+m$ ;  $v=1, \dots, n+m'$ に対して、 $A[\mu, v]$ ;  $\mu=1, \dots, n+m$ ;  $v=1, \dots, n+m'$ を知らずして、等価検証処理における検証式を満たす応答を求めることは、離散対数問題を解くことに等しいからである。

【0281】なぜならば、 $A[\mu, v]$ を知らないと言う事は、少なくとも一つの $g'[\nu, \Gamma]$ に関しては、 $g[\mu,$



$\Gamma$ ;  $\mu=1, \dots, n+m$ を基底とするその表現を知らない。その時、任意の $c$ に関して、検証式を満たす応答を求めることができるなら、 $c[\xi]=1$ 、 $c[v]=0$ ;  $v=0, \dots, \xi-1$ 、 $\xi+1, \dots, n+m$ なる $c[v]$ を選ぶ事で、離散対数を解くことが出来るからである。

【0282】また、挑戦値 $c[v]$ は、コミットメント $g[\mu, \Gamma]$ 、 $g'[\mu, \Gamma]$ を引数に持つため、挑戦値が決定してから、コミットメントを調節することができない(挑戦値生成関数がこの性質を持つことを要求する)。そのため、証明者には、挑戦値はコミットメント決定後に与えられた乱数と考えることができる。

【0283】どの $g[\mu, \Gamma]$ の成分に関して、他の成分を基底とするその表現を知らなければ、検証式を満たす応答を複数作る事は、離散対数問題を解く事に等しい。なぜなら、異なる $r[\mu]$ 、 $r'[\mu]$ に関して、検証式が成り立つならば、両辺を互いに割る事により、 $g[\mu, \Gamma]$ を基底とする、非自明な1の表現が得られる。これは離散対数問題を解く事に等しいからである。

【0284】入力文列生成方法により生成される入力文列  $g[\mu, \Gamma]$ ;  $\mu=1, \dots, n+m$ ;  $\Gamma=0, \dots$ は、いずれかの $\Gamma$ に関してなすベクトル  $g[\mu, \Gamma]$ ;  $\mu=1, \dots, n+m$ が、ハッシュ関数で生成されているか、あるいはハッシュ関数で生成されているベクトルを乗じた等の操作で生成されていることが明らかになっているため、互いに、他を基底として表現する事が計算量的に困難である、と考えられる。

【0285】以上により、検証式を満たす $r[\mu]$ ;  $\mu=1, \dots, n+m$ として、 $g'[\nu, \Gamma]=\prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, \nu]}$  /  $F_p$ ;  $\nu=1, \dots, n+m$ なる $A[\mu, \nu]$ を用いて、 $r[\mu]=\sum_{\nu=1}^{n+m} A[\mu, \nu]c[\nu]$  /  $F_q$ ;  $\mu=1, \dots, n+m$ と生成する以外には、証明者は計算できない。個別公開鍵を用いる方法でも同様である。

【0286】上述のようにある $\Gamma$ に関して  $g'[\nu, \Gamma]=\prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, \nu]}$  /  $F_p$   $\nu=1, \dots, n+m$ の関係が証明されたならば、他の $\Gamma$ に関して以下のように同様に証明される。

【0287】挑戦値生成関数の引数に含めた $g[\mu, \Gamma]$ 、 $g'[\nu, \Gamma]$ に対して検証式が成り立つならば、 $g'[\nu, \Gamma]=\prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, \nu]}$  /  $F_p$   $\nu=1, \dots, n+m$ である。

【0288】なぜならば、

$$\begin{aligned} \sum_{k=1}^n A[h, i]A[h, j]A[h, k] &= \delta'[i, j, k] / F_q \quad i, j, k=1, \dots, n \\ \sum_{k=1}^n (3A[h, 0]A[h, i]A[h, j] + \rho'' \lambda[h]A[h, i]A[h, j]) &= \delta[i, j] \phi[i] / F_q \quad i, j=1, \dots, n \\ \sum_{i=1}^n (3A[h, 0]A[h, 0]A[h, i] + 2\rho'' \lambda[h]A[h, 0]A[h, i]) + \rho' A[0, i] &= \phi[i] / F_q \\ \sum_{i=1}^n (A[h, 0]A[h, 0]A[h, 0] + \rho'' \lambda[h]A[h, 0]A[h, 0]) + \rho' \lambda[0] + \rho' A[0, 0] &= \phi[0] / F_q \end{aligned}$$

$$* g'[\nu, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, \nu]} / F_p \quad \nu=1, \dots, n+m$$

と表した時に検証式が成り立つならば、

$$= \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, \nu]} \{ \sum_{\nu=1}^{n+m} A[\mu, \nu] (A[\mu, \nu] - A'[\mu, \nu]) c[\nu] \} = 1 / F_p$$

が成り立つ。

【0289】ところが、無作為に選ばれた $c[v]$ に関して、これが成り立つのは、

$$\prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, \nu]} = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A'[\mu, \nu]} / F_p \quad \nu=1, \dots, n+m$$

の時だけであるからである。

【0290】前記実施例(2)において、変換条件コミットメント生成処理により準元係数をコミットしたものの $u$ 、 $u[\mu]$ ;  $\mu=0, \dots, n$ が与えられたとき、応答 $r[i]$ ;  $i=1, \dots, n$ と、準応答 $r'$ が検証式を満たす時、準応答 $r'$ は一意であり、

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] / F_q$$

が、検証式を満たすことより $r'$ は上式で表されるものである。

【0291】前記実施例(2)の恒等式の検証式の左辺の $v$ の指数部を展開すると、

$$\begin{aligned} \sum_{k=1}^n \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[h, i]A[h, j]A[h, k]c[i]c[j]c[k] + \sum_{k=1}^n \sum_{i=1}^n \sum_{j=1}^n (3A[h, 0]A[h, i]A[h, j] + \rho'' \lambda[h]A[h, i]A[h, j])c[i]c[j] + \sum_{i=1}^n (\sum_{k=1}^n (3A[h, 0]A[h, 0]A[h, i] + 2\rho'' \lambda[h]A[h, 0]A[h, i]) + \rho' A[0, i])c[i] + \sum_{i=1}^n (A[h, 0]A[h, 0]A[h, 0] + \rho'' \lambda[h]A[h, 0]A[h, 0]) + \rho' \lambda[0] + \rho' A[0, 0] &= \phi[0] / F_q \end{aligned}$$

である。

【0292】右辺の $v$ の指数部は、

$$\sum_{i=1}^n (c[i]c[i]c[i] + \phi[i]c[i]c[i] + \phi[i]c[i]) + \phi[0] / F_q$$

である。

【0293】よって、いかなる $c[\mu]$ ;  $\mu=0, \dots, n$ に関しても、検証式が成り立つためには、 $c[\mu]c[\nu]c[\xi]$ ;  $\mu, \nu, \xi=0, \dots, n$ の係数が同じでなければならない。それ以外の時に、検証式が無作為に与えられた $c[\mu]$ に関して成り立つ可能性は、無視できる。

【0294】これは、

$$\begin{aligned} \delta[i, j] &= 1 \quad i=j \text{ の場合} \\ &= 0 \quad \text{その他} \\ \delta'[i, j, k] &= 1 \quad i=j=k \text{ の場合} \\ &= 0 \quad \text{その他} \end{aligned}$$

\* を用いて、

であることを保証する。このことから、 $A[i, j]; i, j = 1, \dots, n$ について以下のことが分かる。

【0295】与えられた $j, k; j \neq k$ に対して、第 $h$ 成分が $A[h, j]A[h, k]$ である $n$ 次元ベクトル $A[h, j]A[h, k]; h = 1, \dots, n$ と、与えられた $i$ に対して第 $h$ 成分が $A[h, i]$ である $n$ 次元ベクトル $A[h, i]; h = 1, \dots, n$ とを考える。今、 $n$ 個のベクトル $A[h, i]; i = 1, \dots, n$ が $n$ 次元空間を張る、すなわち全てのベクトルが $A[h, i]; i = 1, \dots, n$ の線形結合で表せるとする。すると、上式よりベクトル $A[h, j]A[h, k]; h = 1, \dots, n$ は、すべてのベクトル $A[h, i]$ と内積が0である

$$A[h, j]A[h, k] = 0 \quad /F_q \quad h = 1, \dots, n$$

である。

【0296】このことから、 $n$ 個のベクトル $A[h, i]; h = 1, \dots, n; i = 1, \dots, n$ のうち、各 $h$ 成分が0でないベクトルはひとつしかない。

【0297】また上式より、 $i = j = k$ の時、 $A[h, i]A[h, j]A[h, k] \neq 0$ であるため、ベクトル $A[h, i]; h = 1, \dots, n$ は少なくともひとつは0でない成分を持つ。よって、全てのベクトル $A[h, i]; h = 1, \dots, n$ は、0でない成分をただひとつ持ち、上式より、それは $1^{1/3}$ である。

【0298】今度は、 $n$ 個のベクトル $A[h, i]; h = 1, \dots, n; i = 1, \dots, n$ が $n$ 次元空間を張ることを示す。

【0299】ベクトル $a[h]; h = 1, \dots, n$ を、 $n$ 個のスカラー $\kappa[i]; i = 1, \dots, n$ を用いて、 $a[h] = \sum_{i=1}^n \kappa[i]A[h, i] \quad h = 1, \dots, n \quad /F_q$ と表す。

【0300】もし、 $a[h] = 0 \quad /F_q$ ならば、 $\kappa[i] = 0$ であることを示せば、 $n$ 個のベクトル $A[h, i]; h = 1, \dots, n; i = 1, \dots, n$ が $n$ 次元空間を張ることを示せる。 $a[h] = 0 \quad /F_q$ ならば、上式の両辺に、第 $h$ 成分が $A[h, i]A[h, i]$ である $n$ 次元ベクトル $A[h, i]A[h, i]$ を掛けると、上式の二つ上の式より、

$$0 = \kappa[i] \quad /F_q \quad i = 1, \dots, n$$

となる。以上をもって $A[i, j]$ が置換行列であるか、置換行列のいくつかの成分に $1^{1/3}$ を乗じることによって得られる準置換行列であることが示された。

【0301】実施例(1)の恒等式の検証式の左辺の $v$ の指数部を展開すると、

$$\begin{aligned} & r[0]r[0] + \sum_{i=1}^n r[i]r[i] \quad /F_q \\ & = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[i, j]A[i, k]c[j]c[k] \\ & + \sum_{j=1}^n (\sum_{i=1}^n 2A[i, 0]A[i, j] + r'[0]A[0, j])c[j] \\ & + \sum_{i=1}^n A[i, 0]A[i, 0] + r'[0]A[0, 0] \quad /F_q \end{aligned}$$

である。右辺の $v$ の指数部は、

$$\sum_{i=1}^n (c[i]c[i] + \phi[i]c[i]) + \phi[0] \quad /F_q$$

である。

【0302】よっていかなる $c[\mu]; \mu = 0, \dots, n$ に関しても、検証式が成り立つためには、 $c[\mu]c[v]; \mu, v = 0, \dots, n$ の係数が同じでなければならない。それ以外の時に検証式が無作為に与えられた応答に関して成り立つ

可能性は無視できる。

【0303】これは

$$\sum_{h=1}^n A[h, i]A[h, j] = \delta[i, j] \quad /F_q$$

$$\phi[i] = \sum_{h=1}^n 2A[h, 0]A[h, i] + r'[0]A[0, i] \quad /F_q$$

$$\phi[0] = \sum_{i=1}^n A[i, 0]A[i, 0] + r'[0]A[0, 0] \quad /F_q$$

であることを保証する。よって、 $A[i, j]; i, j = 1, \dots, n$ は正規直交行列でなければ検証式は成り立つ可能性は無視できる。

【0304】前記実施例(3)、実施例(4)においても同様の議論が成り立ち、 $A[i, j]; i, j = 1, \dots, n$ は、置換行列かつ正規直交行列である。そして、これは置換行列であることを意味する。

【0305】[知識隠匿性] 再暗号シャッフル証明文において、再暗号シャッフルの情報が計算量的に秘匿されていることを示す。

【0306】再暗号シャッフルの結果 $g'[v], m'[\mu]$ 以外にも、

$$r[\mu], r', \phi[i], \psi[i], \omega, v', v'', v, u, u[i],$$

$$r[\mu], \phi[i], v', \omega, v$$

等の値が明らかになる。これらは再暗号シャッフルに関する情報を与えている。しかし指数演算された結果でない条件の数より、再暗号シャッフル行列演算に関する未知数のほうが多くなるように、恒等式の係数をコミットして隠蔽すれば離散対数問題をといて条件の数を増やさない限り解けない。ただし変数の数だけでなく未知数の条件の中への現れ方で解ける場合もあるので、若干の調整を行う必要がある。

【0307】

【発明の効果】以上説明したように、本発明によれば、証明付再暗号シャッフルの計算量を、従来の技術と比べて、低減する、という効果を奏する。

【0308】特に、検証処理は事前に計算しておくことができない応用例が多いと考えられるので、検証の計算量を比較すると、入力暗号文の数を $n$ としたとき、従来の技術(1)では、安全変数が160の時乗剰余演算が $320n + 2n$ 回必要とされており、従来の技術(2)では乗剰余演算が $8(n \log n - n + 1)$ 回必要とされていたのに対して、本発明によれば、乗剰余演算は $7n + 14$ 回で済み、 $n > 4$ の時は、いずれの従来の技術よりも乗剰余演算が少ない。

【0309】しかも、本発明においては、検証過程で行う乗剰余は、個別の乗剰余演算ではなく、乗剰余演算の積の計算であることから、個別の乗剰余演算よりも少ない計算量で計算できるため、更なる高速化が望める、という効果を奏する。

【図面の簡単な説明】

【図1】従来の技術1の構成を示す図である。

【図2】従来の技術2の構成を示す図である。

【図3】本発明の実施例における証明付再暗号シャッフル装置と再暗号シャッフル検証装置との情報の入出力を

示す図である。

【図 4】本発明の実施例 1 の証明付再暗号シャッフル装置の詳細を示す図である。

【図 5】本発明の実施例 1 の再暗号シャッフル検証装置の詳細を示す図である。

【図 6】本発明の実施例 2 の証明付再暗号シャッフル装置の詳細を示す図である。

【図 7】本発明の実施例 2 の再暗号シャッフル検証装置の詳細を示す図である。

【図 8】本発明の実施例 3 の証明付再暗号シャッフル装置の詳細を示す図である。

【図 9】本発明の実施例 3 の再暗号シャッフル検証装置の詳細を示す図である。

【図 10】本発明の実施例 4 の証明付再暗号シャッフル装置の詳細を示す図である。

【図 11】本発明の実施例 4 の再暗号シャッフル検証装置の詳細を示す図である。

【図 12】本発明の実施例 5 の入力文列生成装置の詳細を示す図である。

【図 13】本発明の実施例 6 の入力文列生成方法の情報の入出力を示す図である。

【図 14】本発明の実施例 6 における前処理装置の詳細を示す図である。

【図 15】本発明の実施例 7 の入力文列生成方法の情報の入出力を示す図である。

【図 16】本発明の実施例 7 の入力文列生成方法の情報の入出力を示す図である。

【図 17】本発明の実施例 7 における証明付暗号化装置の詳細を示す図である。

【図 18】本発明の実施例 7 における暗号化検証装置の詳細を示す図である。

【図 19】本発明の実施例 6 と実施例 7 における証明付公開鍵列生成装置の詳細を示す図である。

【図 20】本発明の実施例 6 と実施例 7 における公開鍵列検証装置の詳細を示す図である。

【符号の説明】

100 入力暗号文列

101 暗号シャッフル

102 出力暗号文列

103 疑似出力暗号文列

104 挑戦値生成関数

105 挑戦値

106 応答

200 置換

300 入力文列

301 入力暗号文列

303 公開鍵

303 再暗号シャッフル情報

304 再暗号シャッフル行列

305 再暗号秘密乱数

306 情報隠蔽

307 シャッフル行列

308 元係数

309 準元係数

310 係数基底

311 その他

312 証明付き再暗号シャッフル行列装置

313 出力暗号文列

314 暗号シャッフル証明文

315 変換情報保有コミットメント

316 変換条件コミットメント

317 応答

318 準応答

319 再暗号シャッフル検証装置

400、600、800、1000 入力文列

40、601、801、1006 再暗号シャッフル情報

402、602、802、1001 再暗号シャッフル行列

403、603、803、805、1002、1005 元係数

404、604、605、806、1003 係数基底

405、602、808、1009 再暗号シャッフル行列作用

406、603、809、1010 出力文列

407、604、810、1011 出力暗号文列

408、605、811、1012 保有コミット

409、606、812、816、1013、1022 恒等式係数計算

40A、614、823、1017 コミットメント

410、607、813、817、1014、1023 恒等式係数

411、608、814、818、1015、1024 隠蔽処理

412、613、815、822、1016、1028 条件コミット

413、615、824、1030、1034 挑戦値生成

414、616、825、1031、1035 挑戦値

415、617、826、1032、1036 応答生成

416、618、827、1033、1037 応答

417、810、1011 出力暗号文列

418、622、831、1040 再暗号シャッフル証明文

419、623、832、1042 変数情報保有コミットメント生成処理

420、625、834、1043 変換条件コミットメント生成処理

421、624、835、1046、1047 応答生

成処理

500 挑戦値生成関数

501 挑戦値

502、706、902、1103、1105 保有検証

503 条件検証

504、709、906、1110 検証結果

505、710 変換情報保有検証処理

506、711 変換条件検証処理

610、1026 準元係数隠蔽処理

612、821、1027 準応答コミット

619、828 準応答生成

620、829 準応答

60A、807、1007 準元係数

704、900、1100 挑戦値生成

705、901、1101 挑戦値

707、903、904、1102 恒等式検証

708、905、1107 準応答検証

820 準元係数隠蔽処理

821 準応答コミット

1004 情報隠蔽因子

1018 選択処理

1019 入力文列

1020 保有生成

1021 保有コミット

1111、1114 条件検証処理

1112、1113 保有検証処理

1200、1400 基底生成関数

1201 入力ベクトル

1300、1501、1901 公開鍵列情報

1301、1502、1902 秘密鍵

1302、1503、1903 疑似秘密鍵

1304、1504 証明付き公開鍵列装置

1305、1306、1505、1506 分散公開鍵列対

1309 前処理装置

1308 検証結果

1401、1403 公開鍵列基底

1402 前処理

1404 公開鍵列

1500 共通初期値

1507 公開鍵列検証装置

1508 検証結果

150A 共通初期値

1600 共通公開鍵

1601 個別公開鍵

1602 平文

1603 暗号化技術

1604 秘密乱数

1605 疑似秘密乱数

1605 証明付き暗号化装置

1607 入力暗号文

1608 暗号化証明文

1609 暗号化検証装置

1610、1810 検証結果

1700、1701、1702 べき剰余演算

1703 暗号文底

1704 コミットメント

1706、1911 挑戦値生成

1707、1912 挑戦値

1708、1913 応答生成

1709、1914 応答

1710、1904 基底生成関数

1711 公開鍵基底

1712 平文

1713 暗号文対

1800、2000 挑戦値生成関数

1801、2003 挑戦値

1802 暗号化検証

1900 初期値

1904、2000 基底生成関数

1905、2002 公開鍵列底

1906 公開鍵列対生成

1907 公開鍵列対

1908 疑似公開鍵列対生成

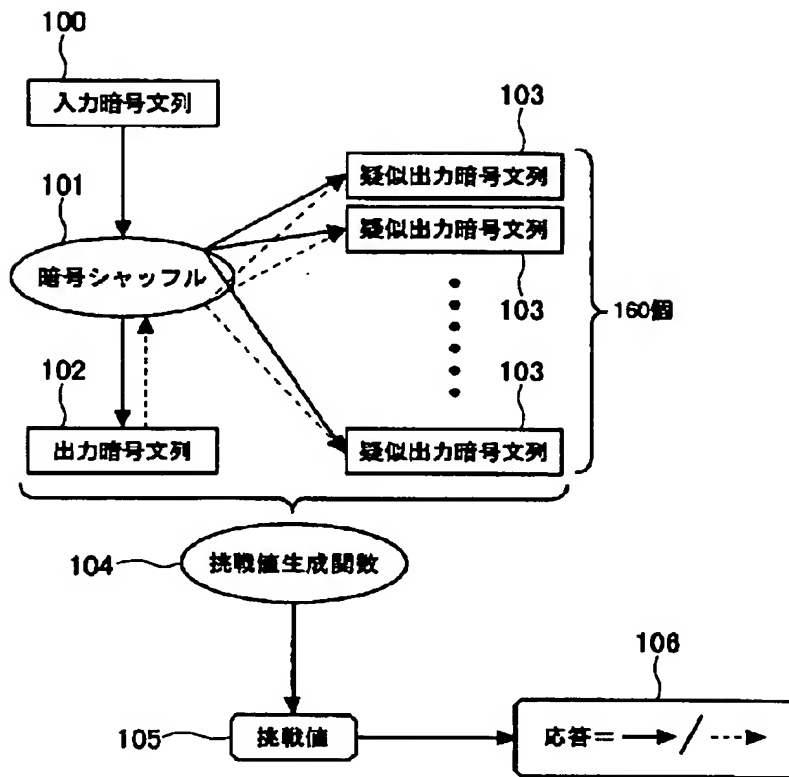
1915 公開鍵列証明文

1917 公開鍵列対

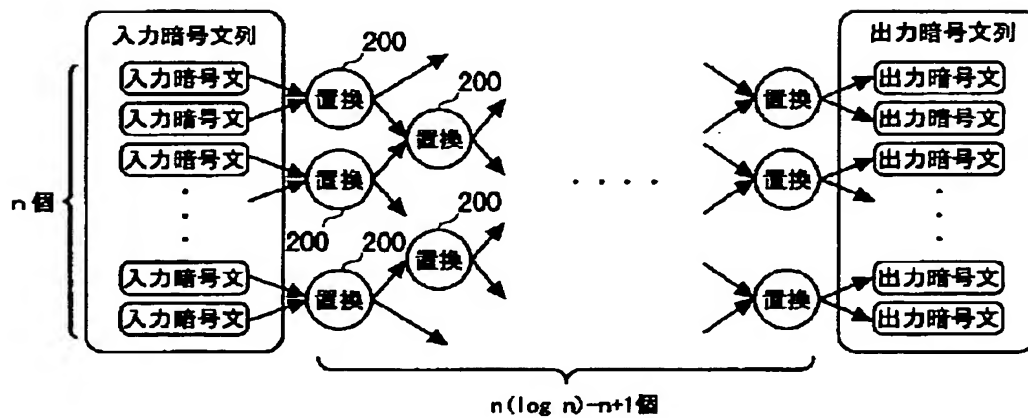
2004 公開鍵列検証

2005 検証結果

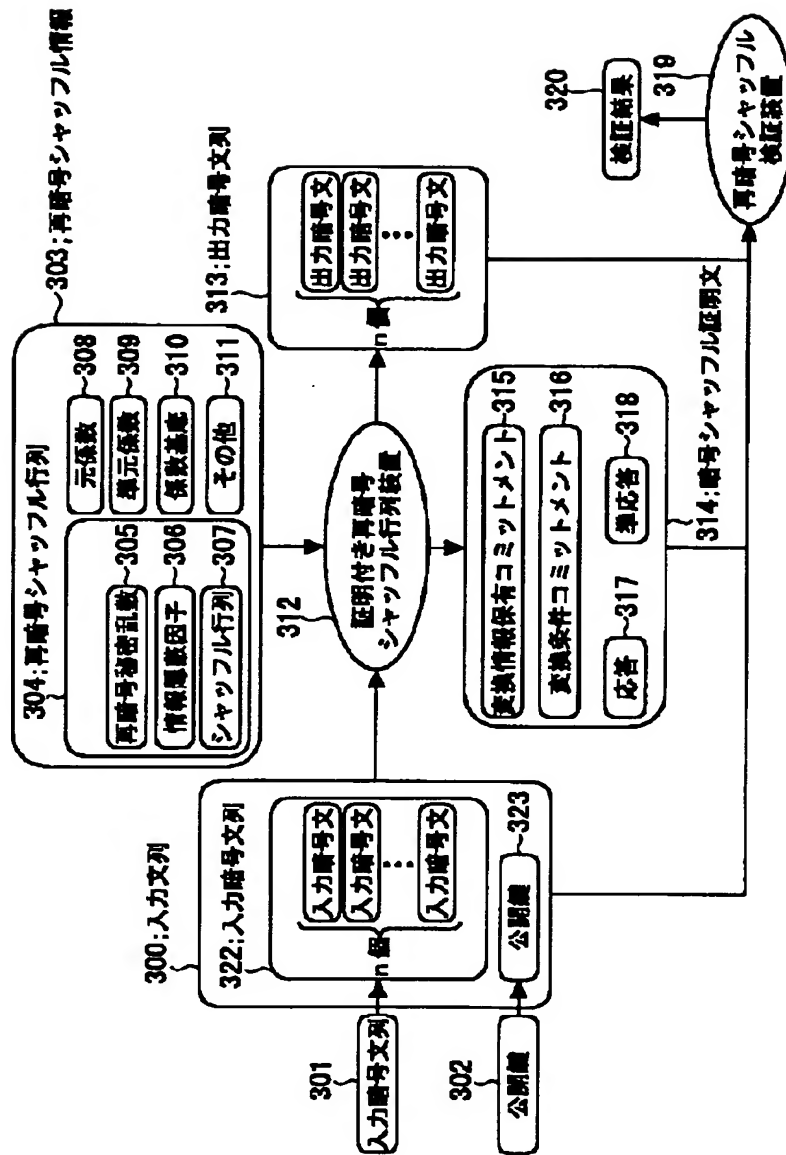
【図1】



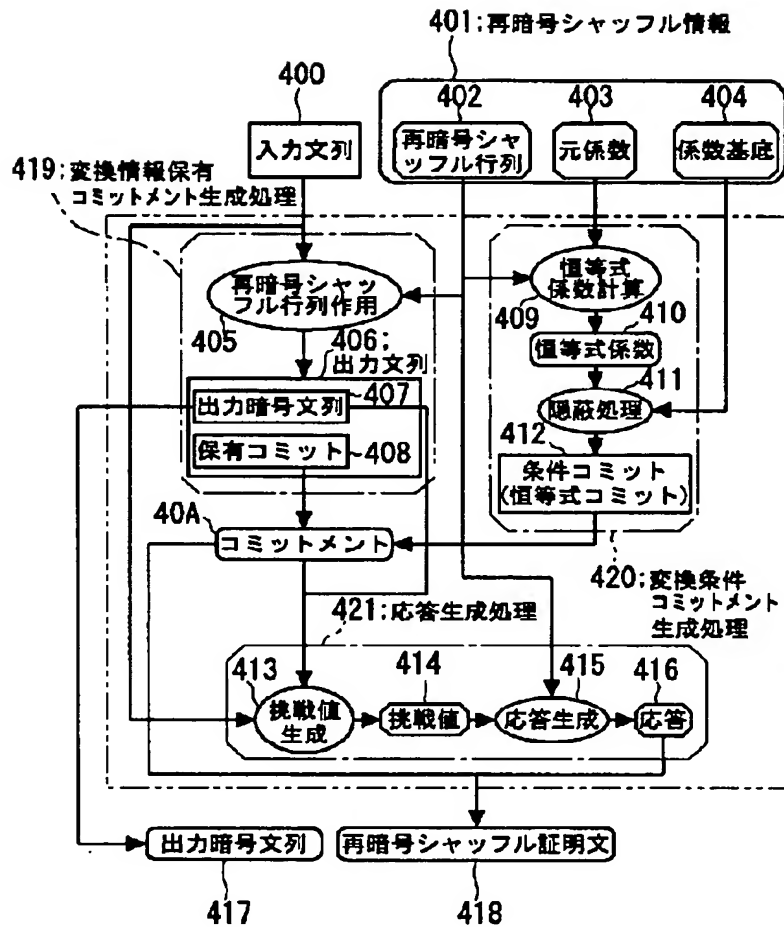
【図2】



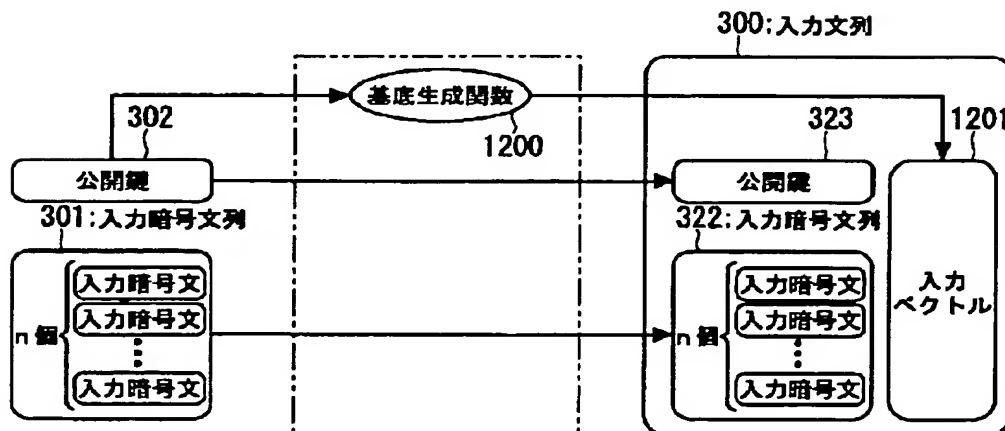
【図3】



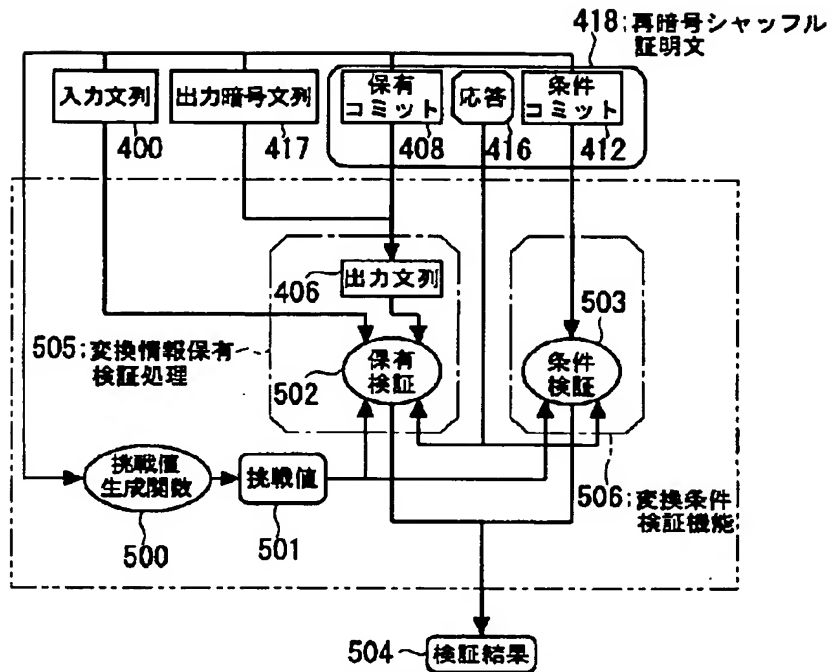
【図4】



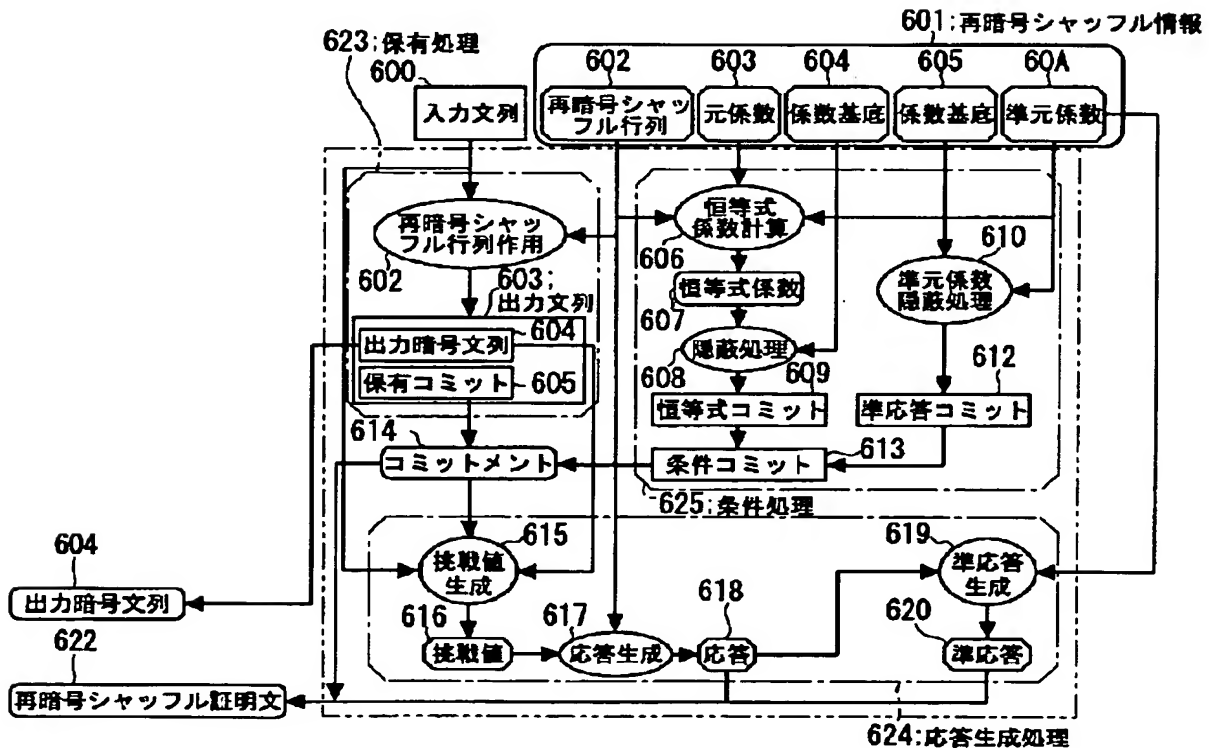
【図12】



【図5】

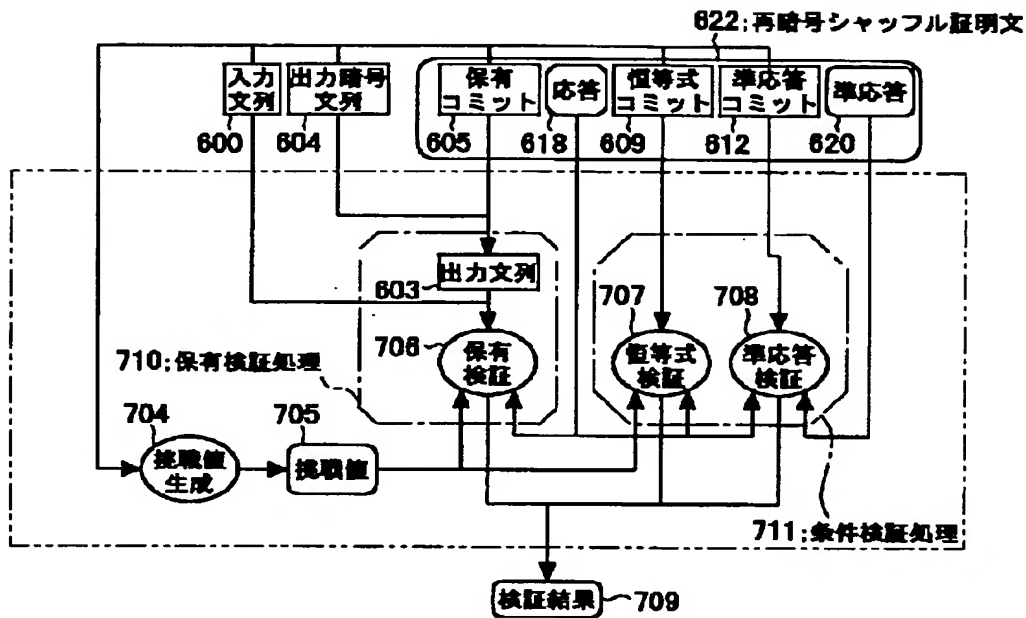


【図6】

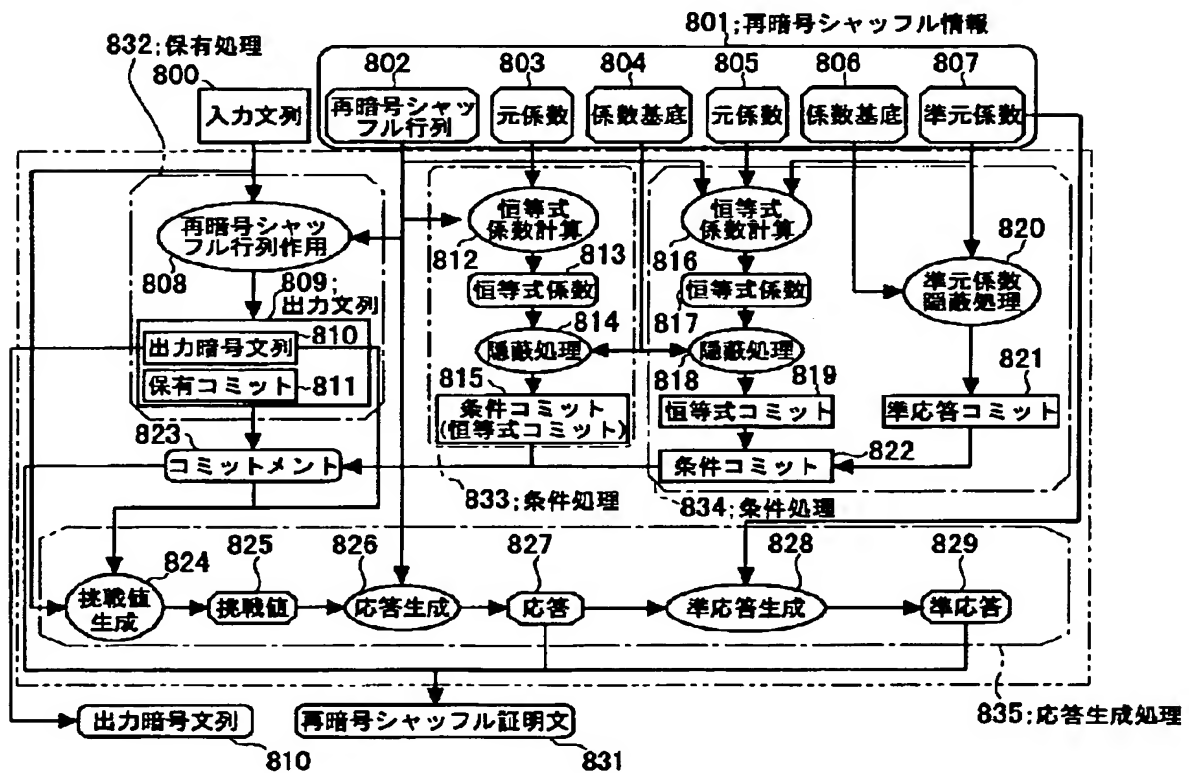




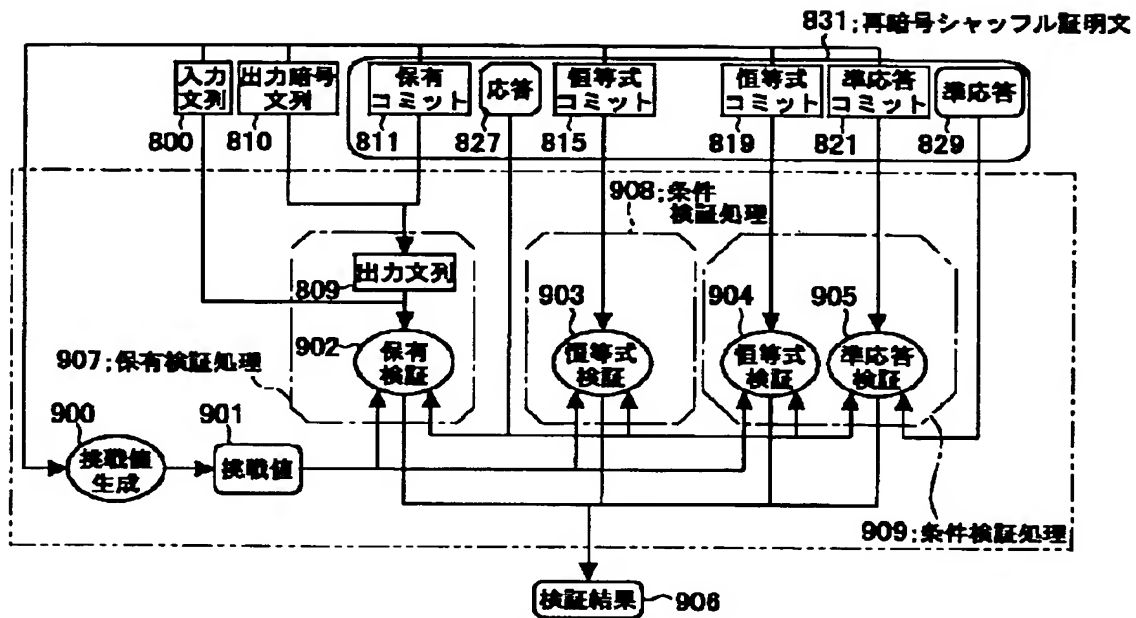
【図7】



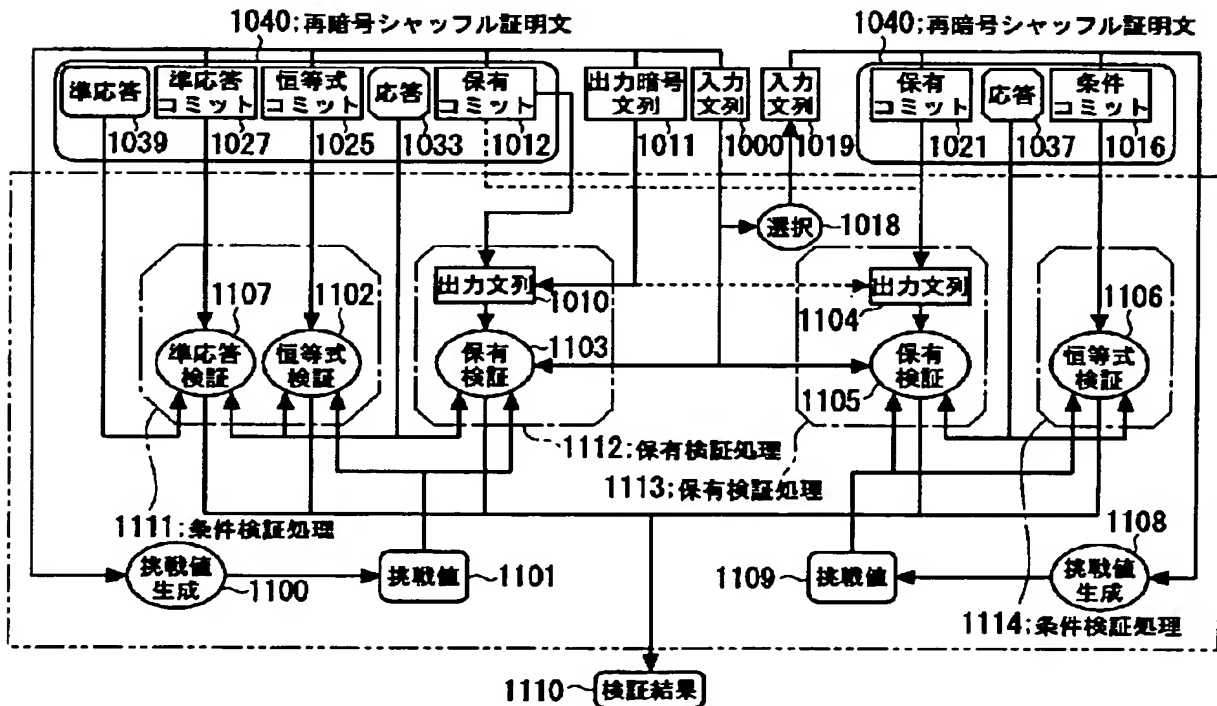
【図8】



【図9】

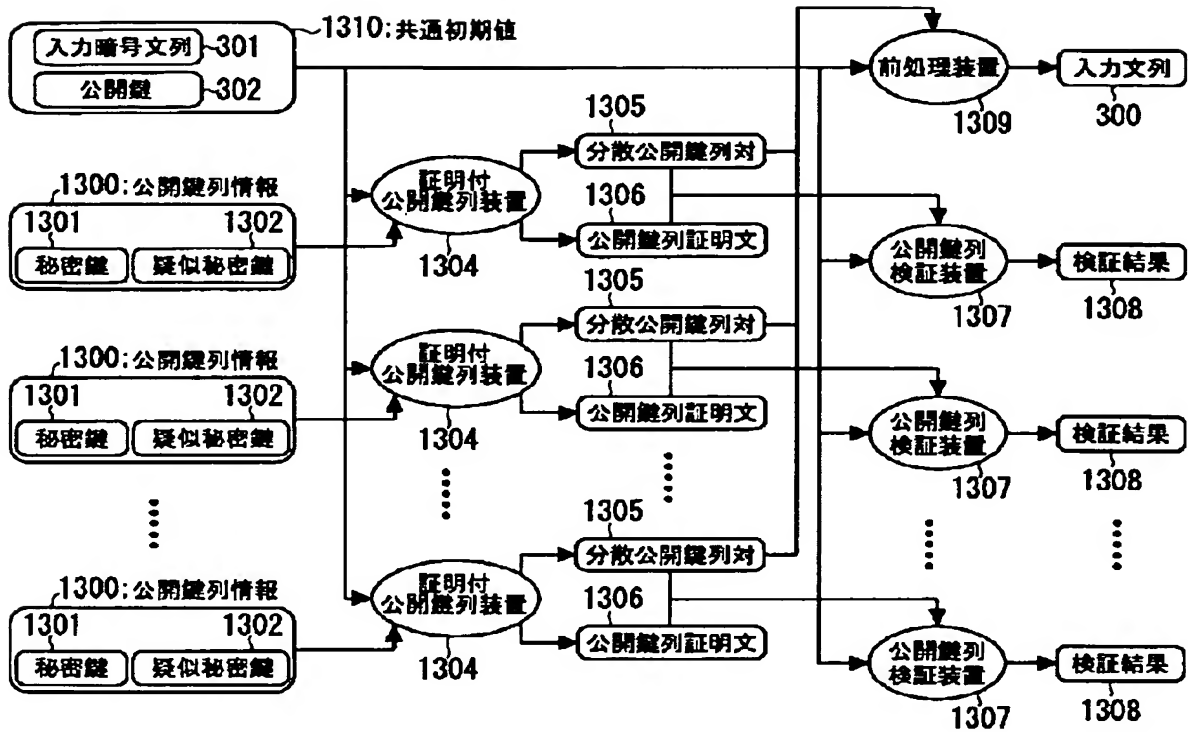


【図11】

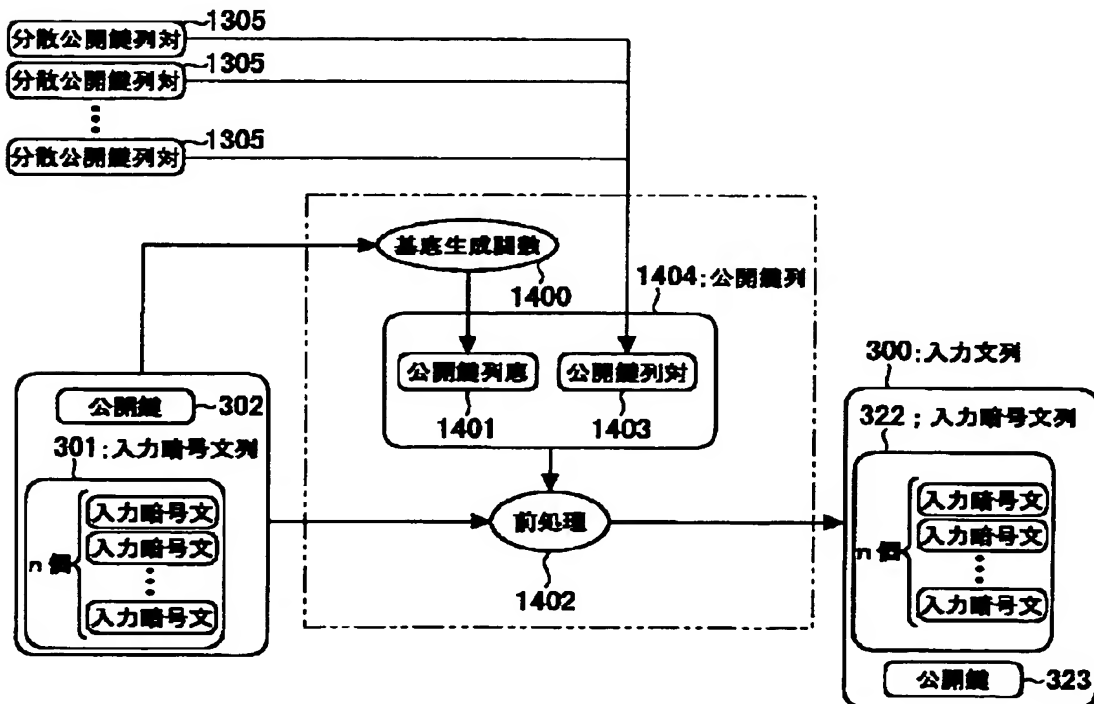


[illegible]

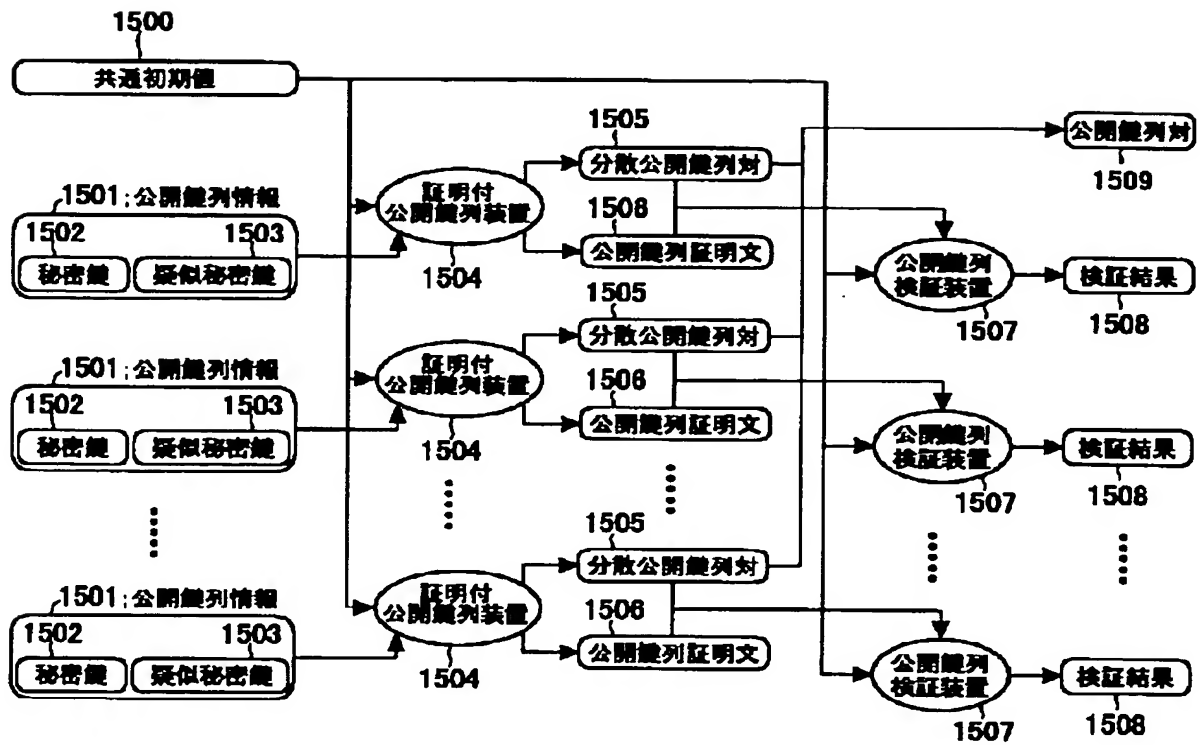
【図13】



【図14】



【図15】



【図18】

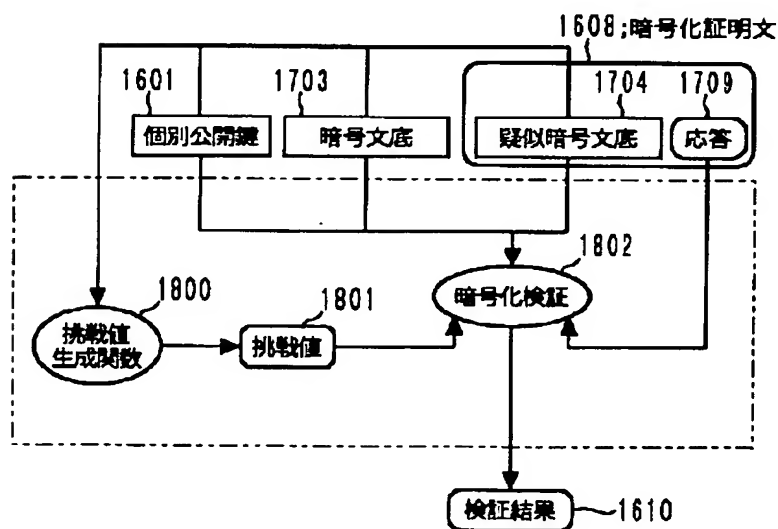
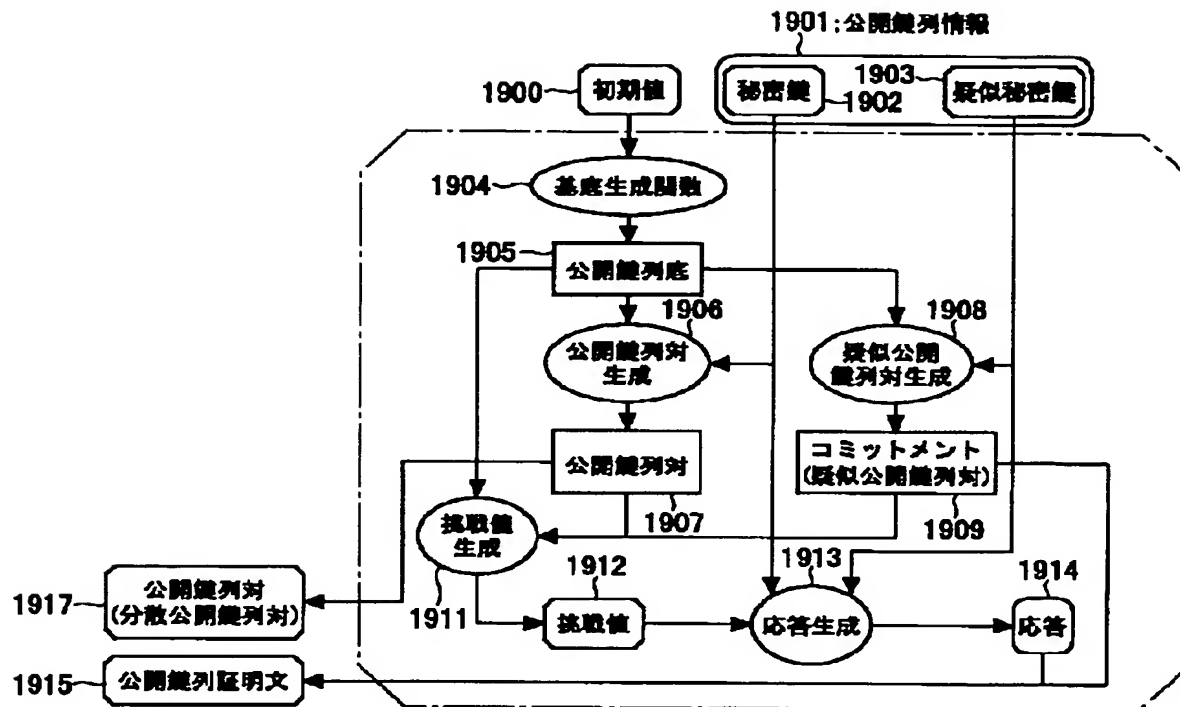


Figure 1 is a block diagram of a cryptographic system 100. The system is composed of multiple parallel processing units, each handling a different type of data (e.g., common key, secret key, plaintext, ciphertext, verification result). The units are interconnected to perform encryption, decryption, and verification operations. The diagram shows a flow from input keys and plaintext through encryption and decryption blocks to produce ciphertext and verification results.

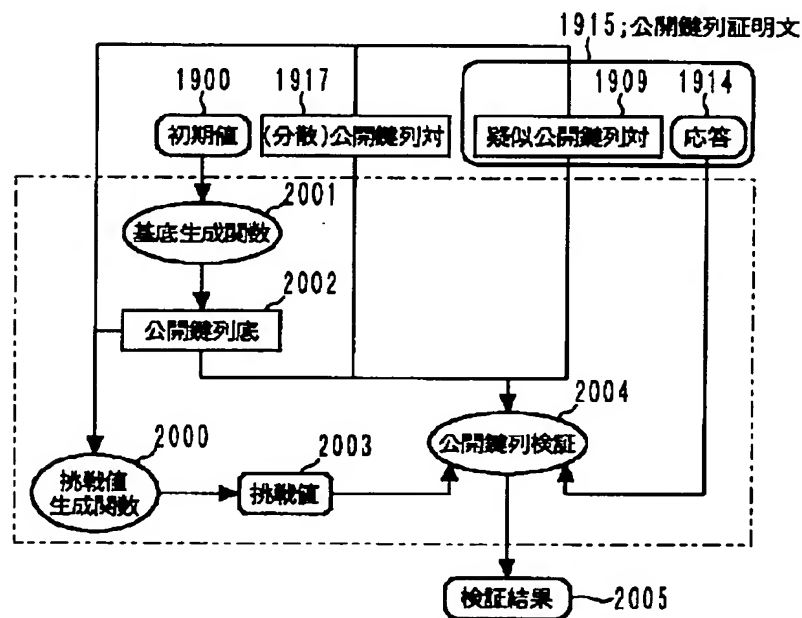




【图 19】



【図 20】



## 【手続補正書】

【提出日】平成12年11月20日(2000. 11. 20)

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0039

【補正方法】変更

【補正内容】

【0039】例えば、入力暗号文列を、 $g[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を、 $g[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ 、それ以外の入力文列の成分を、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$ 、一般の変換に対応する乱数(以下「情報隠蔽因子」と呼ぶ)を、 $A[\mu, j]; \mu=1, \dots, n+m, j=n+1, \dots, n+m'$ 、再暗号化の変数を、 $A[i, j]; i=n+1, \dots, n+m, j=1, \dots, n$ 、並び替えに対応する変換を表す変数を、 $A[i, j]; i, j=1, \dots, n$ として、出力暗号文列  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=1, \dots, l$  を、 $g'[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A[j, i]} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A[j, i]} / F_p, i=1, \dots, n, \Gamma=1, \dots, l$  と生成し、変換情報保有コミットメントを、 $g'[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A[j, i]} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A[j, i]} / F_p, i=n+1, \dots, n+m, \Gamma=1, \dots, l$  と生成し、入力文列に  $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$  が含まれた場合の変換情報保有コミットメントを、 $g'[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A[j, i]} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A[j, i]} / F_p, i=1, \dots, n+m, \Gamma=1+1, \dots, l'$  と生成できる。

## 【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0040

【補正方法】変更

【補正内容】

【0040】また、これらをまとめて  $g'[i, \Gamma] = \prod_{j=1}^{n+m} g[j, \Gamma]^{A[j, i]} / F_p, i=1, \dots, n+m, \Gamma=1, \dots, l'$  と記述できる。

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n c[i]c[i]c[i] + \sum_{i=1}^n \psi[i]c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

が成り立つことを確認し、また準応答の正当性を、検証式

$$u' = u[0] \prod_{i=1}^n u[i]^{r[i]r[i]} / F_p$$

が成り立つことより確認する。

## 【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0074

※

$$\begin{aligned} & \hat{v} \{ \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] \} / F_p \\ & = \hat{v} \{ \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] \} / F_p \end{aligned}$$

## \* 【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0056

【補正方法】変更

【補正内容】

【0056】変換条件コミットメントとして、恒等式の係数または、それをコミットしたものと、準元係数をコミットしたものを生成する。実施例では恒等式の一部を、 $v, v^{(0)} / F_p$  のようにコミットし、準元係数を  $u, u^{(1)} / F_p, \mu=0, \dots, n$  のようにコミットする。

## 【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0068

【補正方法】変更

【補正内容】

【0068】例えば、挑戦値  $c[i]; i=1, \dots, n+m'$  と応答  $r[i]; i=1, \dots, n+m$  が、 $\prod_{i=1}^{n+m} g'[i, \Gamma]^{c[i]} = \prod_{i=1}^{n+m} g[i, \Gamma]^{r[i]} / F_p, \Gamma=1, \dots, l'$  が成り立つことを確認する。

## 【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0073

【補正方法】変更

【補正内容】

【0073】例えば、変換条件コミットメントとして恒等式の係数  $\rho'', \rho'[\mu], \phi[\mu], \psi[i]$  に対して、挑戦値  $c[i]; i=1, \dots, n+m'$  と、応答  $r[i]; i=1, \dots, n+m$  が、恒等式  $\sum_{i=1}^n r[i]r[i]r[i] + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] = \sum_{i=1}^n c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q$  や、恒等式

※ 【補正方法】変更

【補正内容】

【0074】また恒等式の係数の一部がコミットされている場合は、代わりに、 $\hat{v} \{ \sum_{i=1}^n r[i]r[i]r[i] + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] \} / F_p = \hat{v} \{ \sum_{i=1}^n c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] \} / F_p$  や、

$$r[\mu] \} / F_p^*, \\ = v^{\wedge} \{ \sum_{i=1}^n c[i]c[i]c[i] + \sum_{i=1}^n \psi[i]c[i]c[i] + \sum_{i=1}^n \phi[\mu]c[\mu] \} / F_p^*,$$

が成り立つことを確認する。上式で、記号「 $\wedge$ 」は指数演算を示す。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0081

【補正方法】変更

【補正内容】

【0081】例えば、公開鍵列を、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l$ とし、入力暗号文列を、 $\eta[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を、 $\eta[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ としたとき、入力文列  $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l$ を、検証者にも明らかな任意の正整数 $s[i]; i=1, \dots, n+m$ を用いて、 $g[i, \Gamma] = \eta[i, \Gamma] g'[i, \Gamma]^{s[i]} / F_p^*$ と表す。 $s[i]$ として、例えば $s[n+m]=0, s[j]=1; j=1, \dots, n+m-1$ を選ぶ。

【手続補正8】

【補正対象書類名】明細書

【補正対象項目名】0083

【補正方法】変更

【補正内容】

【0083】例えば、公開鍵列を、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=0, 1$ とし、平文を、 $m[i]; i=1, \dots, n; \Gamma=0, 1$ としたとき、入力暗号文を、 $\eta[i, 0] = g'[i, 0]^{s[i]} / F_p^*, i=1, \dots, n$   
 $\eta[i, 1] = m[i] g'[i, 1]^{s[i]} / F_p^*, i=1, \dots, n$ と生成し、合わせて、 $\eta[i, 0] = g'[i, 0]^{s[i]} / F_p^*$ となる $s[i]$ の知識を証明することで、 $g'[i, 0]$ を使って暗号化した証明文とする。

【手続補正9】

【補正対象書類名】明細書

【補正対象項目名】0102

【補正方法】変更

【補正内容】

【0102】挑戦値生成関数の出力は、 $n+1$ 個の1,0でない $q$ 以下の整数、基底生成関数の出力は、 $n+1$ 個の1,0でない $p$ 以下の整数で位数 $q$ の $F_p^*$ の元(位数 $p-1$ の乗法群の位数 $q$ 部分群の元)である整数である。

【手続補正10】

【補正対象書類名】明細書

【補正対象項目名】0105

【補正方法】変更

【補正内容】

【0105】挑戦値生成関数の場合は、 $|q|$ ビットを出力するハッシュ関数を用いて数列を求め、その中から順に1,0でないものを $n+1$ 個選んでいく(この場合、 $k$ 乗する操作は必要無い)。

【公開鍵】公開鍵について説明する。公開鍵は、二つの

数値 $\eta[0, 0], \eta[0, 1]$ であり $\eta[0, 0]$ は位数 $q$ の $F_p^*$ の元とする。 $\eta[0, 1]$ は秘密鍵 $x$ を用いて、

$$\eta[0, 1] = \eta[0, 0]^x / F_p^*$$

と計算される。

【手続補正11】

【補正対象書類名】明細書

【補正対象項目名】0106

【補正方法】変更

【補正内容】

【0106】【入力暗号文】入力暗号文について説明する。平文を、 $p$ 以下で位数 $q$ の $F_p^*$ の元から選び、これを $M$ とする。これから疑似乱数生成器で生成した秘密乱数 $r$ を用いて、入力暗号文を、 $(\eta[0, 0]^r, M \cdot \eta[0, 1]^r) / F_p^*$ と計算する。

【手続補正12】

【補正対象書類名】明細書

【補正対象項目名】0107

【補正方法】変更

【補正内容】

【0107】【再暗号化】再暗号化について説明する。Elgamal暗号文 $(\eta[0, 0]^r, M \cdot \eta[0, 1]^r) / F_p^*$ が与えられた時、任意の乱数 $s$ を選んで、 $(\eta[0, 0]^r, M \cdot \eta[0, 1]^r) \rightarrow (\eta[0, 0]^r \cdot \eta[0, 0]^s, M \cdot \eta[0, 1]^r \cdot \eta[0, 1]^s) / F_p^* = (\eta[0, 0]^{r+s}, M \cdot \eta[0, 1]^{r+s}) / F_p^*$ なる変換を行うことを「再暗号化」という。上記変換は $r$ を知らなくても実行できる。またこの変換により再暗号化された暗号文の復号結果はかわらない。この時の乱数 $s$ を、「再暗号秘密乱数」と呼ぶ。

【手続補正13】

【補正対象書類名】明細書

【補正対象項目名】0110

【補正方法】変更

【補正内容】

【0110】【準置換行列】準置換行列について説明する。「準置換行列」とは、置換行列の1である成分を、 $F_p^*$ 上の3個ある1の3乗根のいずれかで置き換えたものと定義する。これらを $w, w^2, 1$ として下に準置換行列の例をあげる。

【手続補正14】

【補正対象書類名】明細書

【補正対象項目名】0112

【補正方法】変更

【補正内容】

【0112】【再暗号シャッフル】再暗号シャッフルについて説明する。入力暗号文列 $\eta[i, 0], \eta[i, 1]; i=1, \dots, n$ の順序を入れ替えて、暗号文列 $\eta'[i, 0], \eta'[i, 1]; i=1, \dots, n$ を生成し、さらに $n$ 個の秘密乱数 $s[i]; i=$

$1, \dots, n$ と、公開鍵  $\eta[0, 0], \eta[0, 1]$  を用いて、出力暗号文列  $g'[i, \Gamma]; i=1, \dots, n, \Gamma=0, 1$  を、  
 $g'[i, \Gamma] = \eta'[i, \Gamma] \eta[0, \Gamma]^{s(i)} / F_p^*$   $i=1, \dots, n, \Gamma=0, 1$

と計算する。これが、再暗号シャッフルの出力結果である。これを、「出力暗号文列」と呼ぶ。

【手続補正15】

【補正対象書類名】明細書

【補正対象項目名】0115

【補正方法】変更

【補正内容】

【0115】  $g'[\mu, \Gamma] = \Pi_{v=0}^n g[v, \Gamma] A[v, \mu] / F_p^*$   
 $\mu=0, \dots, n, \Gamma=0, 1$

ここで、シャッフル行列が置換行列の場合、出力暗号文列を、 $g'[i, 0], g'[i, 1]; i=1, \dots, n$  とし、展開すると、ある置換  $(i, j) \mid \pi(i)=j$  に対して、

$g'[j, 0] = g[i, 0] \eta[0, 0]^{A[i, j]} / F_p^*$   
 $g'[j, 1] = g[i, 1] \eta[0, 1]^{A[i, j]} / F_p^*$

となり、これは再暗号シャッフルの出力となる。

【手続補正16】

【補正対象書類名】明細書

【補正対象項目名】0116

【補正方法】変更

【補正内容】

【0116】またシャッフル行列が準置換行列の場合、準再暗号シャッフルの結果

$g'[j, 0] = g[i, 0] \eta[0, 0]^{A[i, j]} / F_p^*$   
 $g'[j, 1] = g[i, 1] \eta[0, 1]^{A[i, j]} / F_p^*$

を出力する(準再暗号シャッフルとは各出力暗号文を1または $w$ または $w^2$ 乗すると再暗号シャッフルとなるものと定義する)。ここで  $w[i]; i=1, \dots, n$  は  $F_q$  上の1の三乗根のいずれかをとる。

【手続補正17】

【補正対象書類名】明細書

【補正対象項目名】0122

【補正方法】変更

【補正内容】

【0122】係数基底404  $v$ 、元係数403  $r'[0]$  を生成に関しては、疑似乱数生成器で1, 0でない  $F_q$  上の数を生成し、 $r'[0]$  とし、疑似乱数生成器により  $F_p^*$  の元を生成し  $F_p^*$  上でその  $k$  乗をとり1, 0でないものを選び位数  $q$  の  $F_p^*$  の元を生成し、 $v$  とする。

【手続補正18】

【補正対象書類名】明細書

【補正対象項目名】0123

【補正方法】変更

【補正内容】

【0123】  $r'[0] \in_r F_q, \neq 0, 1$

$v \in_r F_p^*, \neq 1, \text{ s.t. } v^q = 1 / F_p^*$

入力暗号文列  $\eta[i, 0], \eta[i, 1]; i=1, \dots, n$  と、公開鍵  $\eta$

$[0, 0], \eta[0, 1]$  より、入力文列400

$g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$  を、

$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$

$g[i, \Gamma] = \eta[i, \Gamma] / F_p^*, \quad i=1, \dots, n, \Gamma=0, 1$

とする。

【手続補正19】

【補正対象書類名】明細書

【補正対象項目名】0128

【補正方法】変更

【補正内容】

【0128】さらに、係数基底404  $v$  を用いて、隠蔽処理411により、恒等式係数410  $r'[0], \phi[0]$  を、

$v' = v^{r'[0]} / F_p^*$

$\omega = v^{\phi[0]} / F_p^*$

とコミットする。

【手続補正20】

【補正対象書類名】明細書

【補正対象項目名】0136

【補正方法】変更

【補正内容】

【0136】変換条件検証処理506により、挑戦値501と応答416と変換条件コミットメント412とを用いて検証式

$v^{r'[0]} v^{\sum_{i=1}^n r[i] r[i]} = \omega v^{\sum_{i=1}^n (c[i] c[i] + \phi[i] c[i])} / F_p^*$

が成り立つことを確認503する。

【手続補正21】

【補正対象書類名】明細書

【補正対象項目名】0146

【補正方法】変更

【補正内容】

【0146】  $\rho' \in_r F_q, \neq 0, 1$

$\rho'' \in_r F_q, \neq 0, 1$

$v \in_r F_p^*, \neq 1, \text{ s.t. } v^q = 1 / F_p^*$

$\lambda[\mu] \in_r F_q, \neq 0, 1, \mu=0, \dots, n$

$u \in_r F_p^*, \neq 1, \text{ s.t. } u^q = 1 / F_p^*$

【手続補正22】

【補正対象書類名】明細書

【補正対象項目名】0151

【補正方法】変更

【補正内容】

【0151】さらに、係数基底604  $v$  を用いて、隠蔽処理608により、恒等式係数607  $\rho', \rho'', \phi[0]$  を、

$\omega = v^{\phi[0]} / F_p^*$

$v' = v^{\rho'} / F_p^*$

$v'' = v^{\rho''} / F_p^*$

とコミット609する。さらに、係数基底605  $u$  を用いて、準元係数606  $\lambda[\mu]; \mu=0, \dots, n$  を、

$u[0] = u^{\lambda[0]} / F_p^*$

$u[i] = u^{\lambda[i]} / F_p^*, \quad i=1, \dots, n$

とコミット612する。

## 【手続補正23】

【補正対象書類名】明細書

【補正対象項目名】0158

【補正方法】変更

【補正内容】

【0158】変換情報保有検証処理710により、この挑戦値705を用いて、入力文列600と、変換情報保有コミットメント605と、出力暗号文列604である出力文列603

と、応答618と、を用いて検証式、

$$\Pi_{\mu} \circ g[\mu, \Gamma]^{r[\mu]} = \Pi_{\mu} \circ g'[\mu, \Gamma]^{c[\mu]} / F_p^*$$

$$\Gamma = 0, 1$$

が成り立つことを確認706する。

## 【手続補正24】

【補正対象書類名】明細書

【補正対象項目名】0159

【補正方法】変更

【補正内容】

【0159】変換条件検証処理711により、挑戦値705と、応答618と、変換条件コミットメント609、612と、を用いて検証式、

$$v^{r[0]} \cdot v^{\wedge} \{ \sum_{i=1}^n r[i]r[i]r[i] \} = \omega v^{\wedge} \{ \sum_{i=1}^n (c[i]c[i]c[i] + \psi[i]c[i]c[i] + \phi[i]c[i]) \} / F_p^*$$

と検証式707

$$u^{r[1]} = \Pi_{i=1}^n u[i]^{r[i]r[i]} / F_p^*$$

が成り立つことを確認708する。

## 【手続補正25】

【補正対象書類名】明細書

【補正対象項目名】0161

【補正方法】変更

【補正内容】

【0161】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。この

時、出力暗号文列 $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ は、

$$g'[j, 0] = g[i, 0]^{r[i]} g[0, 0]^{A[0, j]} / F_p^*$$

$$g'[j, 1] = g[i, 1]^{r[i]} g[0, 1]^{A[0, j]} / F_p^*$$

を出力した可能性を排除できない。ここで、 $w[i]$ が全て1の時が、再暗号シャッフルである。なお、 $w[i]; i=1, \dots, n$ は、 $F_q$ 上の1の三乗根のいずれかをとり、

## 【手続補正26】

【補正対象書類名】明細書

【補正対象項目名】0168

【補正方法】変更

【補正内容】

【0168】元係数803  $r'[-1], r'[0]$ 、元係数805  $\rho, \rho', \rho''$ 、係数基底804  $v$ 、係数基底806  $u$ 、準元係数807  $\lambda[\mu]; \mu=0, \dots, n$ に関しても、実施例(1)と同様な手法で、 $r'[-1], r'[0], \rho, \rho', \rho'', \lambda[\mu]; \mu=0, \dots, n$ には、1, 0でない $F_q$ 上の数を、係数基底 $u, v$ に

は、位数 $q$ の $F_p^*$ の元を生成する。

## 【手続補正27】

【補正対象書類名】明細書

【補正対象項目名】0169

【補正方法】変更

【補正内容】

$$[0169] r'[-1] \in_r F_q, \neq 0, 1$$

$$r'[0] \in_r F_q, \neq 0, 1$$

$$\rho \in_r F_q, \neq 0, 1$$

$$\rho' \in_r F_q, \neq 0, 1$$

$$\rho'' \in_r F_q, \neq 0, 1$$

$$v \in_r F_p^*, \neq 0, 1, \text{ s.t. } v^q = 1 / F_p^*$$

$$\lambda[\mu] \in_r F_q, \neq 0, 1 \mu = 0, \dots, n$$

$$u \in_r F_p^*, \neq 0, 1, \text{ s.t. } u^q = 1 / F_p^*$$

## 【手続補正28】

【補正対象書類名】明細書

【補正対象項目名】0172

【補正方法】変更

【補正内容】

【0172】変換情報保有コミットメント生成処理832における再暗号シャッフル行列作用808により、上記再暗号シャッフル行列802を入力文列800に以下の様に作用させて、出力文列809 $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$

を、

$$g'[\mu, \Gamma] = \Pi_{v=1}^n g[v, \Gamma]^{A[v \cdot \mu]} / F_p^* \quad \mu = 0, \dots, n, \Gamma = 0, 1$$

と生成する。ここで、 $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ を、出力暗号文列810、 $g'[0, \Gamma]; \Gamma=0, 1$ を、変換情報保有コミットメント811とする。

## 【手続補正29】

【補正対象書類名】明細書

【補正対象項目名】0175

【補正方法】変更

【補正内容】

【0175】さらに係数基底804  $v$ を用いて、隠蔽処理814、818により、恒等式係数813、817  $r'[-1], r'[0], \phi[0], \phi[0], \rho, \rho', \rho''$ を、

$$\omega = v^{r[0]} / F_p^*$$

$$v' = v^{r[0]} / F_p^*$$

$$v' = v^{r[0]} / F_p^*$$

$$\omega' = v^{r[0]} / F_p^*$$

とコミット819し、

$$v = v^{r[-1]} / F_p^*$$

$$v' = v^{r[0]} / F_p^*$$

$$\omega = v^{r[0]} / F_p^*$$

とコミット815する。

## 【手続補正30】

【補正対象書類名】明細書

【補正対象項目名】0176

【補正方法】変更

## 【補正内容】

【0176】さらに、係数基底806  $u$ を用いて、準元係数807  $\lambda[\mu]$ ;  $\mu=0, \dots, n$ を、  
 $u[0]=u^{(0)} / F_p$   
 $u[i]=u^{(i)} / F_p$   $i=1, \dots, n$   
 とコミット821、820する。

## 【手続補正31】

【補正対象書類名】明細書

【補正対象項目名】0179

【補正方法】変更

## 【補正内容】

【0179】応答生成処理835により、以上の入力文列800と、出力暗号文列810と、コミットメント823を、挑戦値生成関数824の引数として、挑戦値825を、  
 $c[0]=1$

$c[i]=\text{Hash}[i](g[\mu, \Gamma], g'[\nu, \Gamma], u[\nu], u, \phi[j], \phi[j], \omega, \omega', \nu', \nu'', \nu, \phi[j], \underline{\omega}, \underline{\nu}, \underline{\nu}; \mu=-1, \dots, n; \nu=0, \dots, n; j=1, \dots, n; \Gamma=0, 1, 2) i=1, \dots, n$   
 と生成し、この挑戦値825から、再暗号シャッフル行列802を用いて、応答827を、  
 $r[\mu]=\sum_{\nu=-1}^n A[\mu, \nu] c[\nu] / F_p$   $\mu=-1, \dots, n$   
 と生成826する。

## 【手続補正32】

【補正対象書類名】明細書

【補正対象項目名】0184

【補正方法】変更

## 【補正内容】

【0184】変換情報保有検証処理907により、この挑戦値901を用いて入力文列800と、変換情報保有コミットメント811と、出力暗号文列810である出力文列809と、応答827とを用いて検証式、  
 $\Pi_{\mu=-1}^n g[\mu, \Gamma]^{r[\mu]} = \Pi_{\nu=0}^n g'[\nu, \Gamma]^{c[\nu]} / F_p$   
 $\Gamma=0, 1$

が成り立つことを確認902する。

## 【手続補正33】

【補正対象書類名】明細書

【補正対象項目名】0185

【補正方法】変更

## 【補正内容】

【0185】変換条件検証処理908、909により、挑戦値901と応答827と準応答829と変換条件コミットメント815、819、821とを用いて、検証式  
 $\hat{v}^{r[\mu]} \hat{v}^{r[0]} \hat{\omega}^{r[-1]} \hat{v} \{ \sum_{i=1}^n r[i] r[i] r[i] \} = \hat{\omega} \hat{v} \{ \sum_{i=1}^n (c[i] c[i] c[i] + \phi[i] c[i] c[i] + \phi[i] c[i]) \} / F_p$

が成り立つことを確認904し、検証式

$\hat{u}^{r[\mu]} = \hat{u}[0] \Pi_{i=1}^n \hat{u}[i]^{r[i] r[i]} / F_p$

が成り立つことを確認905し、検証式

$\hat{\underline{\nu}}^{r[0]} \hat{\underline{\nu}}^{r[-1]} \hat{\underline{\nu}} \{ \sum_{i=1}^n r[i] r[i] \} = \hat{\underline{\omega}} \hat{\underline{\nu}} \{ \sum_{i=1}^n (c[i] c[i] + \phi[i] c[i]) \} / F_p$

が成り立つことを確認903する。

## 【手続補正34】

【補正対象書類名】明細書

【補正対象項目名】0194

【補正方法】変更

## 【補正内容】

【0194】 $\rho' \in_r F_q, \neq 0, 1$

$\rho'' \in_r F_q, \neq 0, 1$

$r'[0], \in_r F_q, \neq 0, 1$

$v \in_r F_p, \neq 1, \text{s.t. } v^q = 1 / F_p$

$\lambda[\mu] \in_r F_q, \neq 0, 1 \mu=0, \dots, n$

$u \in_r F_p, \neq 1, \text{s.t. } u^q = 1 / F_p$

## 【手続補正35】

【補正対象書類名】明細書

【補正対象項目名】0197

【補正方法】変更

## 【補正内容】

【0197】変換情報保有コミットメント生成処理1042における再暗号シャッフル行列作用1009により、上記暗号シャッフル行列1001を入力文列1000に以下の様に作用させて、出力文列1010  $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ を、

$g'[\mu, \Gamma] = \Pi_{\nu=0}^n g[\nu, \Gamma]^{A[\nu, \mu]} / F_p$   $\mu=0, \dots, n, \Gamma=0, 1$

と生成する。

## 【手続補正36】

【補正対象書類名】明細書

【補正対象項目名】0200

【補正方法】変更

## 【補正内容】

【0200】第2の変換情報保有コミットメント1021  $g''[0, \Gamma']$ を、

$g''[0, \Gamma'] = \Pi_{\nu=0}^n g[\nu, \Gamma']^{A[\nu, 0]} / F_p$   $\Gamma'=0 \text{ or } 1$   
 と生成1020する。

## 【手続補正37】

【補正対象書類名】明細書

【補正対象項目名】0202

【補正方法】変更

## 【補正内容】

【0202】さらに、係数基底1003 $v$ を用いて、隠蔽処理1024により、恒等式係数1023  $\phi[0], \rho', \rho''$ を、  
 $\omega = v^{(0)} / F_p$   
 $v' = v^{(0)} / F_p$   
 $v'' = v^{(0)} / F_p$   
 とコミット1025する。

## 【手続補正38】

【補正対象書類名】明細書

【補正対象項目名】0203

【補正方法】変更

## 【補正内容】

【0203】さらに、係数基底1008uを用いて、準元係数1007  $\lambda[\mu]$ ;  $\mu=0, \dots, n$ を、

$$u[0]=u^{(0)} / F_p^*$$

$$u[i]=u^{(i)} / F_p^* \quad i=1, \dots, n$$

とコミット1027する。

【手続補正39】

【補正対象書類名】明細書

【補正対象項目名】0205

【補正方法】変更

【補正内容】

【0205】さらに係数基底1003 vを用いて、隠蔽処理1015により、恒等式係数1014  $r'[0], \phi[0]$ を、

$$v' = v^{(0)} / F_p^*$$

$$\omega = v^{(0)} / F_p^*$$

とコミット1016する。

【手続補正40】

【補正対象書類名】明細書

【補正対象項目名】0215

【補正方法】変更

【補正内容】

【0215】変換情報保有検証処理1112により、第1の挑戦値1101を用いて、入力文列1000と、第1の変換情報保有コミットメント1012と、出力暗号文列1011と、第1の応答1033と、を用いて検証式、

$$\Pi_{\mu=0}^n g[\mu, \Gamma]^{r[\mu]} = \Pi_{\mu=0}^n g'[\mu, \Gamma]^{c[\mu]} / F_p^*,$$

$$\Gamma=0, 1$$

が成り立つことを確認1103する。

【手続補正41】

【補正対象書類名】明細書

【補正対象項目名】0216

【補正方法】変更

【補正内容】

【0216】変換情報保有検証処理1113により、第2の挑戦値1109を用いて第2の入力文列1019と、第2の変換情報保有コミットメント1021と、出力暗号文列1011と、第2の応答1037と、を用いて第2の知識検証式、

$$\Pi_{\mu=0}^n g[\mu, \Gamma']^{r'[\mu]} = g'[0, \Gamma'] \Pi_{i=1}^n g'[i, \Gamma']^{c[i]} / F_p^*, \quad \Gamma'=0$$

が成り立つことを確認1105する。

【手続補正42】

【補正対象書類名】明細書

【補正対象項目名】0217

【補正方法】変更

【補正内容】

【0217】変換条件検証処理1111により、第1の挑戦値1101と、第1の応答1033と、第1の変換条件コミットメント1025とを用いて、検証式1102、

$$v^{(0)} \cdot v^{(0)} \cdot v^{(0)} \{ \sum_{i=1}^n r[i]r[i]r[i] \} = \omega v^{(0)} \{ \sum_{i=1}^n (c[i]c[i]c[i] + \phi[i]c[i]c[i] + \phi[i]c[i]) \} / F_p^*$$

と、準応答1039と、準応答コミットメント1027と、第1

の応答1033と、検証式1107、

$$u^{(0)} = u[0] \Pi_{i=1}^n u[i]^{r[i]r[i]} / F_p^*$$

が成り立つことを確認する。

【手続補正43】

【補正対象書類名】明細書

【補正対象項目名】0218

【補正方法】変更

【補正内容】

【0218】変換条件検証処理1114により、第2の挑戦値1109と、第2の応答1037と、第2の変換条件コミットメント1016とを用いて、検証式1106、

$$v^{(0)} \cdot v^{(0)} \{ \sum_{i=1}^n r[i]r[i] \} = \omega v^{(0)} \{ \sum_{i=1}^n (c[i]c[i] + \phi[i]c[i]) \} / F_p^*$$

が成り立つことを確認する。

【手続補正44】

【補正対象書類名】明細書

【補正対象項目名】0225

【補正方法】変更

【補正内容】

【0225】入力暗号文列301  $\eta[i, 0], \eta[i, 1]; i=1, \dots, n$ と、公開鍵302  $\eta[0, 0], \eta[0, 1]$ が入力され、公開鍵302およびElgamal領域変数p, qより、基底生成関数1200で、入力ベクトル1201を生成し、入力文列300  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1, 2$ を、

$$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$$

$$g[i, \Gamma] = \eta[i, \Gamma] / F_p^* \quad i=1, \dots, n, \quad \Gamma=0, 1$$

$$g[\mu, 2] = \text{Hash}[\mu](p, q, \eta[0, 0], g[0, 1, \Lambda]; \Lambda=1, \dots, t) \mu=0, \dots, n$$

とする。

【手続補正45】

【補正対象書類名】明細書

【補正対象項目名】0226

【補正方法】変更

【補正内容】

【0226】前記実施例(1)から実施例(4)に、本実施例の入力文列生成方法を適用する場合、これらの実施例における $\Gamma$ の値をとる範囲を、全て0、1から、0、1、2に変更する。この新たに導入された、入力暗号文でも公開鍵でもない $\Gamma=2$ の成分が、入力暗号文生成者にも意図できない入力文列の成分となり、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【手続補正46】

【補正対象書類名】明細書

【補正対象項目名】0227

【補正方法】変更

【補正内容】

【0227】また前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力文列を、 $g[-1, \Gamma]$ まで



拡張して、公開鍵  $g[-1, 0]$ 、 $g[-11]$ 、 $\eta[0, 0]$ 、 $\eta[0, 1]$ より、

$$g[-1, \Gamma] = \eta[-1, \Gamma] \quad \Gamma=0, 1$$

$$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$$

$$g[i, \Gamma] = \eta[i, \Gamma] / F_p^* \quad i=1, \dots, n, \Gamma=0, 1$$

$$g[\mu, 2] = \text{Hash}[\mu](p, q, \eta[0, 0], g[0, 1, \Lambda]; \Lambda=1, \dots, t) \quad \mu=-1, \dots, n$$

とする。

【手続補正47】

【補正対象書類名】明細書

【補正対象項目名】0228

【補正方法】変更

【補正内容】

【0228】また実施例(4)に、本実施例の入力文列生成方法を適用する場合、 $\Gamma'=2$ とし、第2の情報隠蔽因子より、第2の変換情報保有コミットメントを、

$$g'_{\Gamma'}[02] = \Pi_{v=-1}^n g[v, 2]^{\Lambda' \cdot v \cdot 0} / F_p^*$$

$$g'_{\Gamma'}[i2] = \Pi_{v=-1}^n g[v, 2]^{\Lambda' \cdot v \cdot i} / F_p^* \quad i=1, \dots, n$$

と変更する。

【手続補正48】

【補正対象書類名】明細書

【補正対象項目名】0230

【補正方法】変更

【補正内容】

【0230】さらに、変換情報保有検証処理における第2の知識検証式は、

$$\Pi_{\mu=0}^n g[\mu, 2]^{\Gamma' \cdot \mu} = \Pi_{\mu=0}^n g'_{\Gamma'}[\mu, \Gamma']^{\Gamma' \cdot \mu} / F_p^*$$

と変更する。

【手続補正49】

【補正対象書類名】明細書

【補正対象項目名】0233

【補正方法】変更

【補正内容】

【0233】公開鍵列検証方法1307により、各証明者の出力した分散公開鍵列対1305と、公開鍵列証明文と、共通初期値1310とから、分散公開鍵列1305の正当性が検証されたら、前処理方法により、各証明者の分散公開鍵列対1305  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n; \Lambda=1, \dots, t$ を合わせて、公開鍵列対140  $3g'[\mu, 1, \Lambda]; \mu=0, \dots, n$ を、 $g'[\mu, 1] = \Pi_{\Lambda=1}^t g'[\mu, 1, \Lambda] / F_p^* \quad \mu=0, \dots, n$ とする。ここで、 $g'[0, 1] = \eta[0, 1]$ に入れ替える。

【手続補正50】

【補正対象書類名】明細書

【補正対象項目名】0236

【補正方法】変更

【補正内容】

【0236】公開鍵列1404と、入力暗号文列301と、公開鍵302とから、入力文列300  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$ を、 $g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$

$g[i, \Gamma] = \eta[i, \Gamma] g'[i, \Gamma] / F_p^* \quad i=1, \dots, n, \Gamma=0, 1$ とする（前処理1402）。

【手続補正51】

【補正対象書類名】明細書

【補正対象項目名】0237

【補正方法】変更

【補正内容】

【0237】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力暗号文列  $\eta[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$ と、公開鍵  $\eta[0, \Gamma]; \Gamma=0, 1$ に対して、公開鍵列  $g'[\mu, \Gamma]; \mu=-1, \dots, n; \Gamma=0, 1$ を生成する。ただし、 $g'[0, \Gamma]; \Gamma=0, 1$ は、公開鍵に等しい。そして、入力文列  $g[\mu, \Gamma]; \mu=-1, \dots, n; \Gamma=0, 1$ を、 $g[-1, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma] = \eta[i, \Gamma] g'[i, \Gamma] / F_p^* \quad i=0, \dots, n, \Gamma=0, 1$ とする。

【手続補正52】

【補正対象書類名】明細書

【補正対象項目名】0242

【補正方法】変更

【補正内容】

【0242】公開鍵列検証方法1507により、各証明者の出力した分散公開鍵列対1505と、公開鍵列証明文1506と、共通初期値1500より、分散公開鍵列対1505の正当性が検証1508されたら、各証明者の分散公開鍵列対1505  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n; \Lambda=1, \dots, t$ を合わせて、公開鍵列対1509  $g'[\mu, 1, \Lambda]; \mu=0, \dots, n$ を、 $g'[\mu, 1] = \Pi_{\Lambda=1}^t g'[\mu, 1, \Lambda] / F_p^* \quad \mu=0, \dots, n$ とする。

【手続補正53】

【補正対象書類名】明細書

【補正対象項目名】0245

【補正方法】変更

【補正内容】

【0245】各入力暗号文生成者  $i=1, \dots, n$ は、証明付暗号化方法1606により、平文1602m[i]と、個別公開鍵1601g[i, \Gamma]; \Gamma=0, 1と、秘密乱数1604s[i]と、疑似秘密乱数1605s'[i]とより、入力暗号文1607  $\eta[i, \Gamma]; \Gamma=0, 1$ を、

$$\eta[i, 0] = g'[i, 0]^{s(i)} / F_p^*$$

$$\eta[i, 1] = m[i] g'[i, 1]^{s(i)} / F_p^*$$

と生成する。

【手続補正54】

【補正対象書類名】明細書

【補正対象項目名】0246

【補正方法】変更

【補正内容】

【0246】またコミットメント(疑似暗号文底1704)、挑戦値1707、応答1709を順に以下のように生成し、 $\eta[i2] = g'[i, 0]^{s(i)} / F_p^*$

$c'[i] = \text{Hash}[0](\eta[i, 0], \eta[i, 1], \eta[i, 2])$

$\theta'[i] = c'[i]s[i] + s'[i] / F_q$

疑似暗号文底1704と応答1709とを暗号化証明文1608とする。

【手続補正55】

【補正対象書類名】明細書

【補正対象項目名】0247

【補正方法】変更

【補正内容】

【0247】暗号化検証装置により、全ての入力暗号文1607と暗号化証明文1608に関して、

$c'[i] = \text{Hash}[0](\eta[i, 0], \eta[i, 1], \eta[i, 2])$

と、挑戦値1801を求め、これに応答1709を用いて、検証式1802

$\eta[i, 0]^{g'[i, 1]} = \eta[i, 1]^{c'[i, 1]} \eta[i, 2] / F_p^*$

が成り立つことを確認1610する。入力暗号文1607全ての正当性が確認されたら、入力暗号文323  $\eta[i, \Gamma]; \Gamma = 0, 1$ と、共有公開鍵1600  $g'[0, \Gamma]; \Gamma = 0, 1$ とより入力文列300を、

$g[0, \Gamma] = g'[0, \Gamma]$

$g[i, \Gamma] = \eta[i, \Gamma] \quad i = 1, \dots, n$

とする。

【手続補正56】

【補正対象書類名】明細書

【補正対象項目名】0254

【補正方法】変更

【補正内容】

【0254】これから、秘密鍵1902  $x$ と、疑似秘密鍵1903  $\alpha$ とにより、(分散)公開鍵列対1907  $g'[\mu, 1]; \mu = 0, \dots, n$ が、

$g'[\mu, 1] = g'[\mu, 0]^x / F_p^*, \quad \mu = 0, \dots, n$

と生成1906され、疑似公開鍵列対1909が、

$g'[\mu, 2] = g'[\mu, 0]^\alpha / F_p^*, \quad \mu = 0, \dots, n$

と生成1908される。

【手続補正57】

【補正対象書類名】明細書

【補正対象項目名】0256

【補正方法】変更

【補正内容】

【0256】公開鍵列検証方法により、挑戦値2003を、

$c'' = \text{Hash}[0](g'[\mu, 0], g'[\mu, 2]; \mu = 0, \dots, n)$

と生成2000し、応答1914を用いて検証式、

$g'[\mu, 0]^{c''} = g'[\mu, 0]^{c'} g'[\mu, 2] / F_p^*, \quad \mu = 0, \dots, n$

を検証2004する。

【手続補正58】

【補正対象書類名】明細書

【補正対象項目名】0263

【補正方法】変更

【補正内容】

$$\prod_{\mu=1}^{n^*} g[\mu, \Gamma]^{r[\mu, \Gamma]} = \prod_{\mu=1}^{n^*} g[\mu, \Gamma]^{\sum_{v=1}^{n^*} A[\mu, v] c[v]}$$

\* 【0263】  $\beta[\Lambda] \in F_q, \neq 0, 1$

【手続補正59】

【補正対象書類名】明細書

【補正対象項目名】0264

【補正方法】変更

【補正内容】

【0264】また、自身の公開鍵  $g[0, 0]$ 、 $g'[0, 1, \Lambda]$  を、 $g[0, 0]$ 、 $g[0, 1]$ と、入力された暗号文列を、 $g[i, \Gamma]; i = 1, \dots, n, \Gamma = 0, 1$ とし、自身の公開鍵と秘密鍵  $x[\Lambda]$  から、部分復号基底  $G[\mu, 0, \Lambda]; \mu = 0, \dots, n$  と、疑似部分復号基底  $G[\mu, 1, \Lambda]; \mu = 0, \dots, n$  を、  
 $G[\mu, 0, \Lambda] = g[\mu, 0]^{x[\Lambda]} / F_p^*, \quad \mu = 0, \dots, n$   
 $G[\mu, 1, \Lambda] = g[\mu, 0]^{g'[0, 1, \Lambda]} / F_p^*, \quad \mu = 0, \dots, n$   
と生成する。コミットメントとして、 $g[\mu, \Gamma, \Lambda]; \mu = 0, \dots, n, \Gamma = 0, 1, \Lambda = 0, \dots, t$  を出力する。

【手続補正60】

【補正対象書類名】明細書

【補正対象項目名】0267

【補正方法】変更

【補正内容】

【0267】部分復号を

$g[i, 0] \rightarrow g[i, 0] \quad i = 1, \dots, n$

$g[i, 1] \rightarrow g[i, 1] / G[i, 0, \Lambda] / F_p^*, \quad i = 1, \dots, n$

として出力する。

【手続補正61】

【補正対象書類名】明細書

【補正対象項目名】0268

【補正方法】変更

【補正内容】

【0268】検証処理は、入力暗号文列と証明文中とより挑戦値を、

$c[\Lambda] = \text{Hash}[0](g[\mu, 0], G[\mu, \Gamma, \Lambda]; \mu = 0, \dots, n; \Gamma = 0, 1)$

と生成し、証明文中の応答、入力暗号文列、分復号基底、疑似部分復号基底を用いて、

$g[\mu, 0]^{c[\Lambda]} = G[\mu, 0, \Lambda]^{c[\Lambda]} G[\mu, 1, \Lambda] / F_p^*, \quad \mu = 0, \dots, n$

を確認し、さらに部分復号が、この  $G[\mu, 0, \Lambda]$  を用いて行われたことを確認して受理する。

【手続補正62】

【補正対象書類名】明細書

【補正対象項目名】0271

【補正方法】変更

【補正内容】

【0271】[完全性] 入力文列と、出力暗号文列と変換情報保有コミットメントとである出力文列と、これに付随する応答と挑戦値が変換情報保有検証処理の検証式を満たすことは、

$$\begin{aligned} & /F_p \\ & = \prod_{v=1}^{n+m} (\prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, v]})^{c[v]} /F_p \\ & = \prod_{v=1}^{n+m} g'[v, \Gamma] /F_p \end{aligned}$$

より分かる。

【手続補正63】

【補正対象書類名】明細書

【補正対象項目名】0272

【補正方法】変更

【補正内容】

【0272】準元係数をコミットしたものと、これに付随する応答と準応答が検証式を満たすことは、

$$\begin{aligned} u' &= u \{ \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] \} /F_p \\ &= (v^{r[0]} \prod_{i=1}^n v^{r[i] r[i]}) /F_p \\ &= (v^{r[0]} v^{\sum_{i=1}^n \sum_{\mu=0}^n \sum_{v=0}^n A[i, \mu] A[i, v] c[\mu] c[v]}) /F_p \\ &= v^{\{ r[0] \sum_{\mu=0}^n [0, \mu] c[\mu] + 2 \sum_{i=1}^n \sum_{j=1}^n A[i, 0] A[i, j] c[j] + \sum_{i=1}^n A[i, 0] A[i, 0] + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[i, j] A[i, k] c[j] c[k] \}} /F_p \\ &= v^{\{ \sum_{i=1}^n \phi[i] c[i] + \phi[0] + \sum_{i=1}^n c[i] c[i] \}} /F_p \\ &= \omega v^{\{ \sum_{i=1}^n (c[i] c[i] + \phi[i] c[i]) \}} /F_p \end{aligned}$$

より成立することがわかる。以上、 $A[i, j]$ が置換行列であるという事実を使った。

【手続補正65】

【補正対象書類名】明細書

【補正対象項目名】0275

$$\begin{aligned} & \sum_{i=1}^n r[i] r[i] r[i] + \sum_{i=1}^n \rho'' \lambda[i] r[i] r[i] + \rho' r[0] /F_p \\ & = \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[h, i] A[h, j] A[h, k] c[i] c[j] c[k] \\ & + \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n (3A[h, 0] A[h, i] A[h, j] + \rho'' \lambda[h] A[h, i] A[h, j]) c[i] c[j] \\ & + \sum_{h=1}^n \sum_{i=1}^n (3A[h, 0] A[h, 0] A[h, i] + 2\rho'' \lambda[h] A[h, 0] A[h, i] + \rho' A[0, i]) c[i] \\ & + \sum_{h=1}^n (A[h, 0] A[h, 0] A[h, 0] + \rho'' \lambda[h] A[h, 0] A[h, 0] + \rho' \lambda[0] + \rho' A[0, 0]) \\ & /F_q \\ & = \sum_{h=1}^n (c[h] c[h] c[h] + \psi[h] c[h] c[h] + \phi[i] c[i] + \phi[0]) /F_q \end{aligned}$$

であり、これは、 $v^{\{ \sum_{h=1}^n (c[h] c[h] c[h] + \psi[h] c[h] c[h] + \phi[i] c[i] + \phi[0]) \}} \omega[0] /F_p$ の指数部と等しい。

【手続補正66】

【補正対象書類名】明細書

【補正対象項目名】0278

【補正方法】変更

【補正内容】

【0278】前記実施例(8)の証明付公開鍵列方法が出力した公開鍵列底と、公開鍵列対と疑似公開鍵列対と、これに付随する応答と挑戦値が検証処理の検証式を満たすことは、

$$\begin{aligned} g'[\mu, 0] &= g'[\mu, 0]^{a''} /F_p \\ &= g'[\mu, 0]^{a''} g'[\mu, 0]^a /F_p \\ &= g'[\mu, 1] g'[\mu, 2] /F_p \end{aligned}$$

から分かる。

【手続補正67】

【補正対象書類名】明細書

【補正対象項目名】0285

【補正方法】変更

【補正内容】

$$\begin{aligned} & * = u^{\lambda[0]} \prod_{i=1}^n (u^{\lambda[i]})^{r[i] r[i]} /F_p \\ & = u[0] \prod_{i=1}^n u[i]^{r[i] r[i]} /F_p \end{aligned}$$

よりわかる。

【手続補正64】

【補正対象書類名】明細書

【補正対象項目名】0274

【補正方法】変更

【補正内容】

【0274】実施例(1)の恒等式係数に関しては、

※【補正方法】変更

【補正内容】

【0275】前記実施例(2)の恒等式係数に関しては、

$$v''^{r[0]} v^{r[0]} \prod_{i=1}^n v^{r[i] r[i]} /F_p$$

※ の $v$ に対する指数部が、

【0285】以上により、検証式を満たす $r[\mu]$ ; $\mu=$

$1, \dots, n+m$ として、 $g'[v, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, v]} /F_p$ ; $v=1, \dots, n+m$ なる $A[\mu, v]$ を用いて、 $r[\mu] = \sum_{v=1}^{n+m} A[\mu, v] c[v] /F_q$ ; $\mu=1, \dots, n+m$ と生成する以外には、証明者は計算できない。個別公開鍵を用いる方法でも同様である。

【手続補正68】

【補正対象書類名】明細書

【補正対象項目名】0286

【補正方法】変更

【補正内容】

【0286】上述のようにある $\Gamma$ に関して

$$g'[v, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{A[\mu, v]} /F_p \quad v=1, \dots, n+m$$

の関係が証明されたならば、他の $\Gamma$ に関しても以下のよう同様に証明される。

【手続補正69】

【補正対象書類名】明細書

【補正対象項目名】0287

【補正方法】変更

## 【補正内容】

【0287】挑戦値生成関数の引数に含めた $g[\mu, \Gamma]$ ,  $g'[\nu, \Gamma]$ に対して検証式が成り立つならば、 $g'[\nu, \Gamma] = \prod_{i=1}^{n+m} g[\mu, \Gamma]^{A'[\mu, \nu]_i} / F_p$ ,  $\nu = 1, \dots, n+m'$ である。

## 【手続補正70】

【補正対象書類名】明細書

【補正対象項目名】0288

【補正方法】変更

## 【補正内容】

【0288】なぜならば、 $g'[\nu, \Gamma] = \prod_{i=1}^{n+m} g[\mu, \Gamma]^{A'[\mu, \nu]_i} / F_p$ ,  $\nu = 1, \dots, n+m'$

\*

\*と表した時に検証式が成り立つならば、

$$= \prod_{i=1}^{n+m} g[\mu, \Gamma]^{A'[\mu, \nu]_i} \cdot (A'[\mu, \nu]_i - A'[\mu, \nu]_i) c[\nu] = 1 / F_p$$

が成り立つ。

## 【手続補正71】

【補正対象書類名】明細書

【補正対象項目名】0289

【補正方法】変更

## 【補正内容】

【0289】ところが、無作為に選ばれた $c[\nu]$ に関して、これが成り立つのは、 $\prod_{i=1}^{n+m} g[\mu, \Gamma]^{A'[\mu, \nu]_i} = \prod_{i=1}^{n+m} g[\mu, \Gamma]^{A'[\mu, \nu]_i} / F_p$ ,  $\nu = 1, \dots, n+m'$ の時だけであるからである。

## 【手続補正書】

【提出日】平成12年12月28日（2000. 12. 28）

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

## 【補正内容】

【書類名】明細書

【発明の名称】証明付再暗号シャッフル方法と装置、再暗号シャッフル検証方法と装置、入力文列生成方法と装置及び記録媒体

## 【特許請求の範囲】

【請求項1】複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から前記出力暗号文列への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成ステップと、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成ステップと、再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、前記再暗号シャッフル情報は、入力暗号文の並び替え方

と、再暗号化に用いた変数と、乱数とを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項2】入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を出力する再暗号シャッフル検証方法において、前記入力文列と、前記出力暗号文列と、前記入力文列から前記出力暗号文列への変換情報の保有に関する変換情報保有コミットメントと、応答と、挑戦値とから、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、前記変換の満たす条件に関する変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、を含み、

前記変換情報保有検証ステップと前記変換条件検証ステップの検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証方法。

【請求項3】請求項1に記載の証明付再暗号シャッフル方法において、

前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、

前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算のみを用いて生成し、前記恒等式の係数、または前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとし、

前記応答生成ステップは、前記応答を、再暗号シャッフル情報と挑戦値から積和演算のみで生成し、  
前記表現は、表現値と基底を対応付けるものであって、無作為に与えられた表現値と基底から、これらを対応付ける表現を計算することは、計算量的に困難とされており、

前記挑戦値は、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分であり、  
前記挑戦値生成関数は、与えられた入力から複数の成分を出力するものであって、それらの出力からは入力を求めること、出力成分間の関係を意図して入力を決定することが、計算量的に困難である関数である、  
ことを特徴とする証明付再暗号シャッフル方法。

【請求項4】請求項2に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、  
前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、ことを特徴とする再暗号シャッフル検証方法。

【請求項5】請求項3に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、  
前記準応答は、再暗号シャッフル検証における、変換情報保有検証処理には使用されないものであり、応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、

前記応答生成ステップは、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、

前記再暗号シャッフル証明文は、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答とを含む、

ことを特徴とする証明付再暗号シャッフル方法。

【請求項6】請求項4に記載の再暗号シャッフル検証方

法において、

前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証方法。

【請求項7】請求項1に記載の証明付再暗号シャッフル方法において、

前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、

前記変換条件コミットメント生成ステップは、複数の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成ステップと、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成ステップと、の両ステップの一方または両方を複数含む、

前記応答生成ステップは、前記応答と、前記変換条件コミットメント生成処理に応じて複数の準応答を生成し、

前記再暗号シャッフル証明文は、複数の前記変換条件コミットメントと、このコミットメントに対応する準応答と、前記応答と、前記変換情報保有コミットメントとを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項8】請求項2に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、複数の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑

戦値の多項式である恒等式が成立することを検証する検証ステップと、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証ステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。

【請求項9】請求項1に記載の証明付再暗号シャッフル方法において、

前記変換情報保有コミットメント生成ステップは、複数個の変換情報保有コミットメント生成ステップよりなり、前記各変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、2番目以降の変換情報保有コミットメント生成処理は、1番目の変換情報保有コミットメント生成ステップと共通する出力の生成を省略し、

前記変換条件コミットメント生成ステップは、複数個の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとするステップと、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成するステップと、の両ステップの一方または両方を複数含む、

前記応答生成ステップは、前記変換情報保有コミットメント生成ステップに応じて複数個の応答を生成し、前記変換条件コミットメント生成ステップに応じて複数個の準応答を生成し、

前記再暗号シャッフル証明文は、前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答とよりなるものである、ことを特徴とする証明付再暗号シャッフル方法。

【請求項10】請求項2に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、複数個の変換情報保

有検証ステップよりなり、前記各変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、複数個の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証するステップと、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証するステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。

【請求項11】請求項3または請求項5に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル方法。

【請求項12】請求項3または請求項5に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものであることを特徴とする証明付再暗号シャッフル方法。

【請求項13】請求項7または請求項9に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項14】証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数、または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として

生成する、ことを特徴とする入力文列生成方法。

【請求項15】請求項14に記載の入力文列生成方法において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする、ことを特徴とする入力文列生成方法。

【請求項16】請求項14に記載の入力文列生成方法において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成方法であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成方法。

【請求項17】請求項14に記載の入力文列生成方法において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する、ことを特徴とする入力文列生成方法。

【請求項18】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する、ことを特徴とする証明付公開鍵列生成方法。

【請求項19】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、を備え、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する、ことを特徴とする証明付再暗号シャッフル装置。

【請求項20】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力

し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、

前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、

前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満たす条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、

を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、

前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、前記変換条件コミットメントと、前記応答と、前記挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証部と、を備え、

前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証装置。

【請求項21】請求項19に記載の証明付再暗号シャッフル装置において、

前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成する手段を備え、

前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力する手段を備え、

前記応答生成部が、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する手段を備えたことを特徴とする証明付再暗号シャッフル装置。

【請求項22】請求項20に記載の再暗号シャッフル検証装置において、

前記変換情報保有検証部が、前記出力暗号文列および変



換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証する手段を備え、前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する手段を備えた、ことを特徴とする再暗号シャッフル検証装置。

【請求項23】請求項21に記載の証明付再暗号シャッフル装置において、前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成する手段と、前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する手段と、を備え、前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、前記応答生成部は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成する手段を備え、前記再暗号シャッフル証明文が、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答よりなる、ことを特徴とする証明付再暗号シャッフル装置。

【請求項24】請求項22に記載の再暗号シャッフル検証装置において、前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証装置。

【請求項25】請求項19に記載の証明付再暗号シャッフル装置において、前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成する手段を備え、前記変換条件コミットメント生成部を複数備え、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミ

ットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成部と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を含み、前記応答生成部が、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数の準応答を生成する手段を備え、前記再暗号シャッフル証明文が、複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントよりなる、ことを特徴とする証明付再暗号シャッフル装置。

【請求項26】請求項20に記載の再暗号シャッフル検証装置において、前記変換情報保有検証部が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、前記変換条件検証部が、複数の変換条件検証部よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証部と、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。

【請求項27】請求項19に記載の証明付再暗号シャッフル装置において、前記変換情報保有コミットメント生成部が、複数の変換情報保有コミットメント生成部よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、2番目以降の変換情報保有コミットメント生成処理は1番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、前記変換条件コミットメント生成部が、複数の変換条件コミットメント生成部よりなり、前記入力文列から前記

出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成部と、

前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を複数含み、

前記応答生成処理が、前記複数の変換情報保有コミットメント生成部の出力に応じて複数の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数の準応答を生成する手段を備え、前記再暗号シャッフル証明文は、前記複数の応答と複数の知識のコミットメントと複数の変換条件コミットメントとそれに対応する準応答よりなる、ことを特徴とする証明付再暗号シャッフル装置。

【請求項28】請求項20に記載の再暗号シャッフル検証装置において、

前記変換情報保有検証部が、複数の変換情報保有検証部よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証部が、複数の変換条件検証部よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証部と、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。

【請求項29】請求項21または請求項23に記載の証明付再暗号シャッフル装置において、

前記変換条件コミットメント生成部における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係

を内包している、ものであることを特徴とする証明付再暗号シャッフル装置。

【請求項30】請求項21または請求項23に記載の証明付再暗号シャッフル装置において、

前記変換条件コミットメント生成部における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものである、ことを特徴とする証明付再暗号シャッフル装置。

【請求項31】請求項25または請求項27に記載の証明付再暗号シャッフル装置において、

前記変換条件コミットメント生成部における複数の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものを含む、ことを特徴とする再暗号シャッフル装置。

【請求項32】請求項19記載の前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成装置。

【請求項33】請求項32に記載の入力文列生成装置において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成装置。

【請求項34】請求項32に記載の入力文列生成装置において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成装置であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成装置。

【請求項35】請求項32に記載の入力文列生成装置において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成装置。

【請求項36】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する手段を備えたことを特徴とする証明付公開

鍵列生成装置。

【請求項37】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する再暗号シャッフル装置であって、

(a) 前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント(「変換情報保有コミットメント」という)を生成する変換情報保有コミットメント生成処理と、

(b) 前記変換の満たす条件に関するコミットメント(「変換条件コミットメント」という)を生成する、変換条件コミットメント生成処理と、

(c) 再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、

(d) 前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、

の前記(a)乃至(d)の処理を証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項38】入力文列と、証明付再暗号シャッフル装置から出力される出力暗号文列と、再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満す条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、

(a) 前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、

(b) 前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、

(c) 前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外是不受理を出力する処理、

の前記(a)乃至(c)の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項39】請求項37に記載の記憶媒体において、前記変換情報保有コミットメント生成処理が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変

数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成し、

前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力し、

前記応答生成処理は、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する、

前記各処理を前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項40】請求項38に記載の記憶媒体において、前記変換情報保有検証処理は、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証処理は、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、

前記各処理を前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項41】請求項37に記載の記憶媒体において、前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、

前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、

前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、

前記応答生成処理は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、前記再暗号シャッフル証明文として、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答を出力する、

前記各処理を前記証明付再暗号シャッフル装置を構成す

るコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 4 2】請求項 4 0 に記載の記録媒体において、前記変換条件検証処理が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、前記変換条件検証処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 4 3】請求項 3 7 に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、複数の前記変換条件コミットメント生成処理を備え、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成処理と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、をからなり、前記応答生成処理は、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数の準応答を生成し、前記再暗号シャッフル証明文として複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントを出力する、前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 4 4】請求項 3 8 に記載の記録媒体において、前記変換情報保有検証処理が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、前記変換条件検証処理が、複数の変換条件検証処理より

なり、

前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証処理と、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証処理と、をからなり、前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項 4 5】請求項 3 7 に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、複数の変換情報保有コミットメント生成処理よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、2 番目以降の変換情報保有コミットメント生成処理は 1 番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、前記変換条件コミットメント生成処理が、複数の変換条件コミットメント生成処理よりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成処理と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、を含み、前記応答生成処理は、前記複数の変換情報保有コミットメント生成部の出力に応じて複数の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数の準応答を生成し、前記再暗号シャッフル証明文として前記複数の応答と複数の知識のコミットメントと複数の変換条件コミットメントとそれに対応する準応答とを出力する、前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項46】請求項38に記載の記録媒体において、前記変換情報保有検証処理が、複数の変換情報保有検証処理よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証処理が、複数の変換条件検証処理よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証処理と、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証処理と、を含み、

前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項47】請求項39または請求項41に記載の記録媒体において、

前記変換条件コミットメント生成処理における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しており、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項48】請求項39または請求項41に記載の記録媒体において、

前記変換条件コミットメント生成処理における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しており、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項49】請求項43または請求項45に記載の記録媒体において、

前記変換条件コミットメント生成処理における複数の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項

の各3乗の和と、挑戦値の成分の一部の各3乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含み、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項50】請求項37に記載の記録媒体において、前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項51】請求項50に記載の記録媒体において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項52】請求項50に記載の記録媒体において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成処理であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵であり、前記各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数とする、入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項53】請求項50に記載の記録媒体において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項54】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する証明付公開鍵列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、匿名通信路の構成などに使われる、入出力暗号文の一对一の対応関係を秘匿しつつ一对一の対応の存在を保証する再暗号シャッフル技術および再暗号シャッフル検証技術に関する。

【0002】

【従来の技術】〔従来の技術(1)〕従来の証明付再暗号シャッフルの技術について、例えば特開平08-263575号



公報（「文献1」という）の記載が参照される。図1に、該文献1に記載される構成を示す。なお、本願添付図面中で、合流する矢印は、矢印の元の情報が、全て集まって矢印の先へ送られることを意味し、分岐する矢印は矢印の元の情報全てまたは一部が、それぞれの矢印の先へ送られることを意味する。また破線は使用する入力文列生成法に依存することを示す。

【0003】図1において、160個ある疑似出力暗号文列は零知識証明のコミットメントである。挑戦値は、入出力暗号文とコミットメントより生成して、応答は挑戦値のビット値に応じて実線または点線の矢印で示される入力暗号文列または出力暗号文列から疑似出力暗号文列への写像の明示である。

【0004】図1に示すように、複数のElgamal入力暗号100の順序を並び替えて再び暗号化して出力する手法が紹介されている。このような処理を、「暗号シャッフル」という。この処理が正当であることを保証する為に同文献では以下の手法が紹介されている。並べ替えと再暗号の秘密乱数を毎回異なるものにして、再暗号シャッフルと同様の操作を安全変数(約160)の回数繰り返して疑似出力暗号文列を出力し、これを正当性証明のコミットメントとする。そしてこれら入出力暗号文とコミットメントのハッシュ値を挑戦値105として生成する。

【0005】この挑戦値のビット列を上から順に読み、ビットが“1”の時は入力暗号文列から、“0”の時は出力暗号文列からの並び替え(並び替えを表す写像)と再暗号化(再び暗号化した時に使ったの乱数)の明示を応答106とする。

【0006】以上のコミットメント、挑戦値、応答を再暗号シャッフルの証明文として出力する。以上においてハッシュ値のビット値に応じて対応関係を明示する方法をCut&Choose(カット・アンド・チューズ)法と言う。

【0007】〔従来の技術(2)〕他の従来技術としては、阿部が、1999年電子情報通信学会情報セキュリティ技術報告書で発表した、「AMix-network on Permutation Networks」(「文献2」という)が参照される。この文献2では、例えば図2に示すように、一対の入力暗号文の置換200を繰り返して、全体として複数の入力暗号文の並び替えを実現する。各入力暗号文の置換の証明を、Cut&Choose法でない方法で構成することにより、ある数より小さい入力暗号文数の証明付再暗号シャッフルとしては効率の向上を達成している。すなわち、個々の入力暗号文を置換することによって入力暗号文列全体の並び替えを実現している。個々の置換の証明は効率の良いものであるが、置換を多くそろえる必要がある。

【0008】

【発明が解決しようとする課題】しかしながら、上記した従来の技術は下記記載の問題点を有している。

【0009】従来の技術(1)においては、コミットメントの生成のために安全変数(約160)の回数暗号シャッ

フルしなければならない。一回の再暗号シャッフルは、入力暗号文の数の2倍の冪乗剰余演算を行わねばならず、計算量が多い。

【0010】また検証は、コミットメントを生成するのと同数の冪乗剰余演算を行わねばならず、計算量が多い。

【0011】次に従来の技術(2)においては、一対の入力暗号文の置換とその証明のコミットメントは、合わせて、8回の冪乗剰余演算が必要である。

【0012】この一置換当りの計算量は、従来の技術(1)の2入力暗号文あたりの計算量(=320)と比較すると小さいものの、入力暗号文全体のどのような並び替えでも実現できる回数にわたる一対の入力暗号文の置換が必要とされており、この数は、入力暗号文の数を $n$ とすると、 $n \log n - n + 1$ である。

【0013】このため、入力暗号文の数が増大すると、計算量が多くなる。

【0014】したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、入力暗号文数に依存せずに証明の計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。

【0015】本発明の他の目的は、検証処理を、証明同様に計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。これ以外の本発明の目的、特徴、利点等は以下の実施の形態の記載から、当業者には直ちに明らかとされるであろう。

【0016】

【課題を解決するための手段】前記目的を達成する本発明による証明付再暗号シャッフル方法は、複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント(「変換情報保有コミットメント」という)を生成する変換情報保有コミットメント生成ステップと、前記変換の満たす条件に関するコミットメント(「変換条件コミットメント」という)を生成する、変換条件コミットメント生成ステップと、再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とからなる、ことを特徴とする。

【0017】また本発明に係る再暗号シャッフル検証方法は、入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を

出力する再暗号シャッフル検証方法において、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、を含み、前記変換情報保有検証処理と前記変換条件検証処理の検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする。

【0018】本発明の証明付再暗号シャッフル装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、を備え、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する。

【0019】本発明の再暗号シャッフル検証装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、前記再暗号シャッフル装置から出力される前記出力暗号文列と、前記再暗号シャッフル装置から出力される前記再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証部と、を備え、前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不

受理を出力する。

【0020】また本発明に係る入力文列生成方法は、証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する。本発明においては、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列としてもよい。

【0021】本発明に係る記録媒体は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、

（a）前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成処理と、（b）前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成処理と、（c）再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、（d）前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、の前記（a）乃至（d）の処理を再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録している。

【0022】本発明に係る記録媒体は、入力文列と、前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換の満たす条件に関する変換条件コミットメントと、前記応答とからなる再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、（a）前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、（b）前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、（c）前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する処理、の前記（a）乃至

（c）の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録している。

【0023】〔発明の概要〕本発明は、再暗号シャッフル



ルを、より一般的な変換の一種として表現し、この変換の情報を保有していることの証明と、この変換の満たす条件の証明との二つを合わせて、再暗号シャッフルの証明を構成している。

【0024】これら二種の証明それぞれは、従来の再暗号シャッフルの証明よりも単純であり、入力暗号文数に依存せずに、証明の計算量が短縮されたものであり、また、これらを合わせた再暗号シャッフルの証明でも、この優位性が保たれる。

【0025】変換の情報を保有していることの証明は、出力暗号文列と変換情報保有コミットメントを生成した後、挑戦値から上記変換と、変換情報保有コミットメント生成に使用する乱数とに依存して、応答を生成することにより行う。

【0026】ここで、応答と挑戦値の関係に変換が反映されるため、変換の満たす条件から挑戦値に依存せずに、応答と挑戦値の満たす関係式が存在する。この関係式をコミットして変換の満たす条件を証明する。

【0027】証明すべき変換の満たす条件として、再暗号シャッフルを表わす変換の満たす条件を選べば、両証明をもってして、再暗号シャッフルの証明を構成することができる。

【0028】

【発明の実施の形態】本発明の上記および他の目的、特徴および利点を明確にすべく、以下添付した図面を参照しながら本発明の実施の形態につき詳細に説明する。

【0029】最初に前提となる事柄について述べる。本発明において用いられる暗号方法は、確率暗号である公開鍵暗号系に属する暗号方法である。例えば、Elgamal暗号、楕円暗号、代数曲線暗号などもこれに含まれる。

【0030】本発明に係る証明付再暗号シャッフル方法は、入力暗号文の作成者全員が証明付再暗号シャッフルを行う証明者に対して、入力暗号文の制作に用いた秘密変数を漏らさなければ、証明者は再暗号シャッフル証明文を偽造できないものである。ただし、本発明に係る入力文列生成方法を併せて用いることで、入力暗号文の作成者が証明者と共謀した場合においても、証明文の偽造を防止する、ことができる。

【0031】本発明に係る証明付再暗号シャッフル方法は、変換情報保有コミットメントを生成する変換情報保有コミットメント生成処理と、変換条件コミットメントを生成する変換条件コミットメント生成処理と、応答や準応答を生成する応答生成処理とよりなり、証明文はこの上記3種類の処理より生成されるコミットメントと応答(応答および準応答)とよりなる。

【0032】本発明に係る再暗号シャッフル検証方法は、入力文列と出力暗号文列と変換情報保有コミットメントと応答とから変換の情報を保有していることを検証する変換情報保有検証処理と、変換条件コミットメントと応答と準応答とから変換の満たす条件を検証する変換

条件検証処理とよりなる。

【0033】[変換情報保有コミットメント生成処理]証明付再暗号シャッフル方法を構成する変換情報保有コミットメント生成処理について説明する。

【0034】変換情報保有コミットメント生成処理は、入力文列から再暗号シャッフルに対応する変換を行って出力暗号文列を生成し、入力文列から乱数による一般の変換を行って変換情報保有コミットメントを生成する。

【0035】また入力文列に入力暗号文列と公開鍵以外が含まれている場合には、それから再暗号シャッフルに対応する変換を行ったものも変換情報保有コミットメントとする。

【0036】応答を複数個生成する場合には、異なる乱数による一般の変換を、複数個行い、変換情報保有コミットメントを、複数組生成する。

【0037】例えば、この変換を、出力暗号文列および変換情報保有コミットメントを、入力文列を基底として、再暗号化に用いた変数および乱数および並び替えに対応する値とを表現とした表現値として生成できる。

【0038】また、この表現とは、基底とある表現値を対応付けるものであり、かつ、基底と表現値から表現を計算することが計算量的に困難となる方法である必要があり、この表現方法に、冪乗剰余を用いることができる。

【0039】例えば、入力暗号文列を、 $g[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を、 $g[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ 、それ以外を入力文列の成分を、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$ 、一般の変換に対応する乱数(以下「情報隠蔽因子」と呼ぶ)を、 $A[\mu, j]; \mu=1, \dots, n+m, j=n+1, \dots, n+m'$ 、再暗号化の変数を、 $A[i, j]; i=n+1, \dots, n+m, j=1, \dots, n$ 、並び替えに対応する変換を表す変数を、 $A[i, j]; i, j=1, \dots, n$ として、出力暗号文列  $g''[i, \Gamma]; i=1, \dots, n; \Gamma=1, \dots, l$  を、 $g''[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A[i, j]} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A[i, j]} / F^*$ ,  $i=1, \dots, n; \Gamma=1, \dots, l$

と生成し、変換情報保有コミットメントを、

$$g''[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A[i, j]} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A[i, j]} / F^*, \quad i=n+1, \dots, n+m'; \Gamma=1, \dots, l$$

と生成し、入力文列に $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1+1, \dots, l'$ が含まれた場合の変換情報保有コミットメントを、

$$g''[i, \Gamma] = \prod_{j=1}^n g[j, \Gamma]^{A[i, j]} \prod_{j=n+1}^{n+m} g[j, \Gamma]^{A[i, j]} / F^*, \quad i=1, \dots, n+m'; \Gamma=1+1, \dots, l'$$

と生成できる。

【0040】また、これらをまとめて

$$g''[i, \Gamma] = \prod_{j=1}^{n+m} g[j, \Gamma]^{A[i, j]} / F^*, \quad i=1, \dots, n+m'; \Gamma=1, \dots, l'$$

と記述できる。

【0041】この時、 $g''[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l$ を「出力文列」と呼ぶ。ここで、 $g''[\mu, \Gamma]$ が

表現値で、 $A[\mu, v]$ が表現で、 $g[\mu, \Gamma]$ が基底である。

【0042】変換情報保有コミットメントを応答の数に応じて、複数組生成する場合には、異なる $A[\mu, j]; \mu = 1, \dots, n+m, j = n+1, \dots, n+m'$ を複数用意して生成する。

【0043】この変換情報保有コミットメントと、入力文列と出力暗号文列および挑戦値に対応する応答を証明者が検証式を満たすように生成できることが、入力文列から出力暗号文列への変換の知識を有していることの証明となる。

【0044】[変換条件コミットメント生成処理]証明付再暗号シャッフル方法を構成する変換条件コミットメント生成処理について説明する。

【0045】応答と挑戦値の関係に、入力文列から出力文列および変換情報保有コミットメントへの変換の満たす条件が反映される。そのため、挑戦値に依存せずに成り立つ、応答と挑戦値の関係式が存在する。この関係式をこの変換の満たす条件を表現するものとしてコミットしたものを変換条件コミットメントとする。

【0046】応答を複数個生成する場合は、知識の隠蔽因子の違いを関係式に反映させる。例えばこの関係式を、応答と挑戦値の多項式である恒等式とし、その係数をコミットするか、あるいはこの多項式の一部の項を準応答として、準応答の係数をコミットして変換条件コミットメントとできる。また挑戦値決定後に応答と準応答を生成すればよい。

【0047】応答の各成分は挑戦値の多項式であるが、一部のこの多項式の、一部の項の各2乗の和と、挑戦値の成分の一部の各2乗の和とが、挑戦値によらずに等しくなる関係を内包しているものや、上記にて各3乗の和とが、挑戦値によらずに等しくなる関係を内包している恒等式を実施例では用いている。

【0048】これに対応する実施例における恒等式は、挑戦値 $c[i]$ 、応答 $r[i]$ を用いて、

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i] + \rho'' r' + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n r[i]r[i]r[i] + \rho'' (\lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i]) + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n c[i]c[i]c[i] + \sum_{i=1}^n \phi[i]c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

があげられる。

【0054】ここでは、変換の満たす条件に対応する関係式を内包するように恒等式の係数 $\rho''$ 、 $\rho'[i]$ 、 $\phi[\mu]$ 、 $\phi[i]$ を求めなければならない。

【0055】また恒等式の一部

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i] / F_q$$

を準応答として、準元係数 $\lambda[\mu]; \mu = 0, \dots, n$ をコミットする場合もある。

【0056】変換条件コミットメントとして、恒等式の係数または、それをコミットしたものと、準元係数をコミットしたものを生成する。実施例では恒等式の一部を、 $v, v^{\circ 3} / F_p$

$$\begin{aligned} & * \sum_{i=1}^n (\sum_{j=1}^n A[i, j]c[j])^2 = \sum_{i=1}^n c[i]^2 / F_q \\ & \text{や、} \\ & \sum_{i=1}^n (\sum_{j=1}^n A[i, j]c[j])^3 = \sum_{i=1}^n c[i]^3 / F_q \end{aligned}$$

の関係を含んだものを用いている。

【0049】なお、

$$\begin{aligned} & \sum_{j=1}^n A[i, j]c[j] / F_q \quad i=1, \dots, n \\ & \text{は、} r[i] \text{を構成する挑戦値の多項式} \\ & \sum_{j=1}^{n+m} A[i, j]c[j] / F_q \quad i=1, \dots, n \\ & \text{の一部である。} \end{aligned}$$

【0050】例えばこれらの関係式は、入力文列から出力暗号文列および変換情報保有コミットメントへの変換を定義している変数 $A[\mu, v]; \mu = 0, \dots, n+m; v = 0, \dots, n+m'$ における $A[i, j]; i, j = 0, \dots, n$ が、正規直交行列であることや、準置換行列であることの性質を反映した関係式である。

【0051】「置換行列」とは、正方向行列で、どの行どの列にもただ一つだけ0でない成分が存在し、その値が1である行列のことである。正規直交行列でありかつ準置換行列である行列は置換行列である。

【0052】「準置換行列」とは、上記置換行列の1である成分を、1の3乗根のいずれかで置き換えたものとする。ただし、それぞれの成分毎に異なる1の3乗根で置き換えを行っても良い。この時、置換行列に対応する変換は再暗号シャッフルに対応している。すなわち、この変換条件コミットメント生成処理により変換の満たす条件を証明することによって変換が再暗号シャッフルであることを証明できる効果がある。

【0053】例えば、上記関係式を内包した恒等式の例として、

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i] + \sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & = \sum_{i=1}^n c[i]c[i] + \sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

や、

のようにコミットし、準元係数を $u, u^{1/F_p} / F_p, \mu = 0, \dots, n$ のようにコミットする。

【0057】恒等式の係数をコミットすることと、準応答を用いることには、検証者が応答とコミットメントから再暗号シャッフルを特定するための情報を減じる効果がある。

【0058】[応答生成処理]証明付再暗号シャッフル方法を構成する応答生成処理について説明する。

【0059】応答生成処理では、変換情報保有コミットメントと変換条件コミットメントと入力文列と出力暗号文列を挑戦値生成関数に入力して挑戦値をえる。

【0060】ここで、「挑戦値生成関数」とは、出力が

ら入力を求めることや、異なる出力成分間の関係を意図して入力を決定することが計算量的に困難である関数である。これにより挑戦値が入力とコミットメントと出力とが決定後に、証明者の意図を入れずに生成されたことが保証できる。

【0061】挑戦値生成関数を用いない場合には、検証者が、入力と出力とコミットメントとが示された後に無作為に選ぶことで挑戦値を得る。

【0062】挑戦値から、再暗号シャッフル方法と情報隠蔽因子とを反映した応答や準応答を生成する。

【0063】応答や準応答を複数個生成する場合は、各応答は異なる情報隠蔽因子を反映させる。

【0064】例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする表現値が等しくなるような応答を生成すればよい。

【0065】例えば、応答 $r[\mu]$ ;  $\mu=1, n+m$ は、挑戦値 $c[\mu]$ ;  $\mu=1, \dots, n+m$ を用いて、 $r[\mu]=\sum_{v=1}^{n+m} A[\mu, v]c[v] / F_q$   $\mu=1, \dots, n+m$ を、準応答として、 $r'=\lambda[0]+\sum_{i=1}^n \lambda[i]r[i]r[i] / F_q$ を生成する。

【0066】[変換情報保有検証処理] 再暗号シャッフル検証方法を構成する変換情報保有検証処理について説明する。

【0067】入力文列と出力暗号文列と変換情報保有コミットメント間の関係を、応答と挑戦値の関係が反映していることを検証する。例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする\*

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i]+\rho''r'+\sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & =\sum_{i=1}^n r[i]r[i]r[i]+\rho''(\lambda[0]+\sum_{i=1}^n \lambda[i]r[i]r[i])+\sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu] / F_q \\ & =\sum_{i=1}^n c[i]c[i]c[i]+\sum_{i=1}^n \psi[i]c[i]c[i]+\sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q \end{aligned}$$

が成り立つことを確認し、また準応答の正当性を、検証式

$$u' = u[0] \prod_{i=1}^n u[i]^{r(i)r(i)} / F_p$$

が成り立つことより確認する。

【0074】また恒等式の係数の一部がコミットされて※

$$\begin{aligned} & v\{\sum_{i=1}^n r[i]r[i]r[i]+\rho''r'+\sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu]\} / F_p \\ & =v\{\sum_{i=1}^n r[i]r[i]r[i]+\rho''(\lambda[0]+\sum_{i=1}^n \lambda[i]r[i]r[i])+\sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu]\} / F_p \\ & =v\{\sum_{i=1}^n c[i]c[i]c[i]+\sum_{i=1}^n \psi[i]c[i]c[i]+\sum_{\mu=1}^{n+m} \phi[\mu]c[\mu]\} / F_p \end{aligned}$$

が成り立つことを確認する。上式で、記号「 $\wedge$ 」は指数演算を示す。

【0075】[入力文列生成方法] 本発明に係る証明付再暗号シャッフル方法は、応答と挑戦値の關係に、入力文列から出力暗号文列と変換情報保有コミットメントへの変換を反映する必要があった。そのためには、挑戦値が与えられたときに生成できる応答が制限されていなく

\* 表現値が等しくなるような応答と挑戦値の關係が存在することを確認する。

【0068】例えば、挑戦値 $c[i]$ ;  $i=1, \dots, n+m$ と応答 $r[i]$ ;  $i=1, \dots, n+m$ が、 $\prod_{i=1}^{n+m} g'[i, \Gamma]^{c(i)} = \prod_{i=1}^{n+m} g[i, \Gamma]^{r(i)} / F_p$   $\Gamma=1, \dots, l$ が成り立つことを確認する。

【0069】挑戦値は、証明文生成に用いた値と同じ値を用いる。これは、挑戦値生成関数を用いる場合は、挑戦値生成関数への入力が証明文と入出力に存在するので可能である。

【0070】[変換条件検証処理] 再暗号シャッフル検証方法を構成する変換条件検証処理について説明する。

【0071】変換条件コミットメントより、挑戦値と応答が変換の満たす条件を反映した關係を満たしていることを検証する。

【0072】例えば、変換の満たす条件を反映した關係を内包する恒等式に、応答と挑戦値または、応答と挑戦値と準応答を代入して、恒等式が成立することを確認している。また準応答がある場合、応答と準応答と準元係数をコミットしたものより、準応答の正当性も確認する。

【0073】例えば、変換条件コミットメントとして恒等式の係数 $\rho''$ ,  $\rho'[\mu]$ ,  $\phi[\mu]$ ,  $\psi[i]$ に対して、挑戦値 $c[i]$ ;  $i=1, \dots, n+m$ と、応答 $r[i]$ ;  $i=1, \dots, n+m$ が、恒等式

$$\sum_{i=1}^n r[i]r[i]r[i]+\sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu]=\sum_{i=1}^n c[i]c[i]+\sum_{\mu=1}^{n+m} \phi[\mu]c[\mu] / F_q$$

や、恒等式

※いる場合は、代わりに、

$$\begin{aligned} & v\{\sum_{i=1}^n r[i]r[i]r[i]+\sum_{\mu=1}^{n+m} \rho'[\mu]r[\mu]\} / F_p \\ & =v\{\sum_{i=1}^n c[i]c[i]c[i]+\sum_{\mu=1}^{n+m} \phi[\mu]c[\mu]\} / F_p \end{aligned}$$

や、

ればならない。しかし、入力暗号文の生成情報を証明者が知っている場合この制限を破れる可能性がある。そしてこれを阻止するための方法が入力文列生成方法である。

【0076】本発明に係る入力文列生成方法は、疑似乱数を生成し、これにより入力文列を変換するか、この疑似乱数を入力文列に加えることにより、入力暗号文の生

成者にさえ入力文列を決定できない入力文列を生成する。

【0077】【入力文列生成方法(1)】疑似乱数を生成して、入力暗号文列と公開鍵にくわえて入力文列とする。この時疑似乱数を決められた入力より決定して再現性を保証する。

【0078】例えば、入力暗号文列を、 $g[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を $g[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ としたとき、決まった入力から疑似乱数を、 $(n+m) \times (l'-1)$  ;  $l'-1 \geq 1$ 個生成し、これを、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=l+1, \dots, l'$ として、入力文列を、 $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l'$ とする。

【0079】【入力文列生成方法(2)】入力暗号文列を構成する各暗号文と公開鍵を、入力暗号文列と公開鍵を入力として生成した公開鍵列を構成する各公開鍵で再暗号化して、これをあわせて入力文列とする。

【0080】ここで「公開鍵列」とは、入力から疑似乱数を公開鍵列を構成する公開鍵の数と同数一意的に生成し、この各乱数が各公開鍵のいずれかの要素となるようにしたものを用いる。

【0081】例えば、公開鍵列を、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=1, \dots, l$ とし、入力暗号文列を、 $\eta[i, \Gamma]; i=1, \dots, n; \Gamma=0, \dots, l$ 、公開鍵を、 $\eta[i, \Gamma]; i=n+1, \dots, n+m; \Gamma=0, \dots, l$ としたとき、入力文列  $g[i, \Gamma]; i=1, \dots, n+m; \Gamma=l+1, \dots, l$ を、検証者にも明らかな任意の正整数 $s[i]; i=1, \dots, n+m$ を用いて、 $g[i, \Gamma]=\eta[i, \Gamma]g'[i, \Gamma]^{s[i]} / F^*$ と表す。 $s[i]$ として、例えば $s[n+m]=0, s[j]=1; j=1, \dots, n+m-1$ を選ぶ。

【0082】【入力文列生成方法(3)】各入力平文を公開鍵列を構成する対応する各公開鍵を用いて暗号化し、この公開鍵を用いて暗号化したことの証明をする。この証明文が受理された暗号文と、公開鍵をあわせて入力文列とする。

【0083】例えば、公開鍵列を、 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=0, 1$ とし、平文を、 $m[i]; i=1, \dots, n; \Gamma=0, 1$ としたとき、入力暗号文を、 $\eta[i, 0]=g'[i, 0]^{s[i]} / F^*, i=1, \dots, n$   
 $\eta[i, 1]=m[i]g'[i, 1]^{s[i]} / F^*, i=1, \dots, n$ と生成し、合わせて、 $\eta[i, 0]=g'[i, 0]^{s[i]} / F^*$ となる $s[i]$ の知識を証明することで、 $g'[i, 0]$ を使って暗号化した証明文とする。

【0084】この証明文が検証された暗号文より、入力文列を

$g[i, \Gamma]=\eta[i, \Gamma] \quad i=1, \dots, n; \Gamma=0, 1$   
 $g[i, \Gamma]=g'[i, \Gamma] \quad i=n+1, \dots, n+m; \Gamma=0, 1$ とする。

【0085】【証明付公開鍵列生成方法】与えられた入力から一意的に疑似乱数列を発生させ、その各乱数から

与えられた手続きにより作られた値を成分に持ち、かつ同じ秘密鍵を持つ公開鍵を、各乱数に対応して複数個生成する。これと同時に、全ての公開鍵が同じ秘密鍵を持つことの証明文を生成する。

【0086】複数人で秘密鍵を分散所持している場合は、各人が各秘密鍵で分散して公開鍵列を生成した後、それらを合わせて統合した公開鍵列を生成する。

【0087】例えば、疑似乱数生成器Hash(\*)が与えられ、入力\*から出力を得る。そして出力をまた入力することを繰り返すことにより、再帰的に、疑似乱数列を生成する。この数列の各値を $k$ 乗してできた数列から0, 1を除いて得られる $n+m$ 個の値よりなる数列 $g'[i, 0]; i=1, \dots, n+m$ の各値を成分に持ち、同じ秘密鍵を持つ公開鍵 $g'[i, \Gamma]; i=1, \dots, n+m; \Gamma=0, \dots, l$ を各乱数に対応して生成する。

【0088】ここで秘密鍵を $x[\Gamma]; \Gamma=1, \dots, l$ とすると、公開鍵列は、 $g'[i, 0]=g'[i, 0]$   
 $g'[i, \Gamma]=g'[i, 0]^{x[\Gamma]} / F^*, i=1, \dots, n+m; \Gamma=1, \dots, l$ と表せる。

【0089】上記公開鍵列を正しく生成したことの証明文を生成する。

【0090】秘密鍵を分散所持している場合、各自で分散秘密鍵に対応する公開鍵列を生成し、最後にそれらを合わせて秘密鍵に対する公開鍵列を生成する。

【0091】

【実施例】本発明の実施例について図面を参照して説明する。以下の実施例では、Elgamal暗号を使った例に即して説明する。図中では、略語が使用されており、例えば保有コミットとは変換情報保有コミットメント、条件コミットとは変換条件コミットメント、恒等式コミットとは恒等式係数のコミットメント、準応答コミットとは準応答の係数のコミットメント、保有処理とは変換情報保有コミットメント生成処理、条件処理とは変換条件コミットメント生成処理、応答処理とは応答生成処理、保有検証処理とは変換情報保有検証処理、条件検証処理とは変換条件検証処理のことをいう。

【0092】図3は、本発明に係る証明付再暗号シャッフル装置および再暗号シャッフル検証装置の実施例における入出力について示す図である。

【0093】図3を参照すると、本発明の一実施例においては、複数の入力暗号文322と公開鍵323からなる入力文列300と、並び替え方を決めるシャッフル行列307と、再暗号化の変数である再暗号秘密乱数305と、変換情報保有コミットメントを生成するための乱数である情報隠蔽因子306と、からなる再暗号シャッフル行列304と、恒等式の係数の種となる元係数308と、恒等式の一部である準応答319の係数である準元係数309と、これらをコミットするための係数基底310とよりなる変換条件コミットメントを生成するための諸定数と、を含む再暗号シャ

ッフル情報303と、が、証明付再暗号シャッフル装置312に入力され、出力暗号文列313と再暗号シャッフル証明文314が出力される。

【0094】再暗号シャッフル証明文314は、恒等式の係数、または、それをコミットしたものと準応答の係数をコミットしたものとを含む変換条件コミットメント316と、変換情報保有コミットメント315と、応答317と、準応答318とを含む。

【0095】これら入力文列300と、出力暗号文列313と、再暗号シャッフル証明文314とが、再暗号シャッフル検証装置319に入力され、受理または不受理の検証結果320が出力される。

【0096】上記証明付再暗号シャッフル装置は、証明者が入力暗号文列の生成情報を知らない時のみに、再暗号シャッフル証明文を偽造できない。この偽造をいかなる場合にも阻止するために加える方法が、入力文列生成方法であり、3種類の入力文列生成方法の実施例を挙げる。また、このうち二つの入力文列生成方法で使用される証明付公開鍵列生成方法についても説明する。

【0097】以下では、証明付再暗号シャッフル方法と、入力文列生成方法と、証明付個別公開鍵列生成方法に共通して、前提となる事項から順に説明していく。

【0098】[Elgamal領域変数] まずElgamal領域変数について説明する。

【0099】この変数は、二つの素数 $p, q$ であり、これらは、 $p=kq+1$

なる関係を満たす。ここで、 $k$ は整数である。

【0100】[挑戦値生成関数と基底生成関数] 挑戦値生成関数と基底生成関数について説明する。これらは順に、

$\text{Hash}[\mu; \mu=0, \dots, n](*)$ ,  $\text{Hash}'[\mu; \mu=0, \dots, n](*)$ とする。

【0101】両関数の添字についているギリシャ文字 $\mu$ は0から $n$ までの値をとり、引数「\*」を入力すると、それぞれ $n+1$ 成分のベクトルを出力する。

【0102】挑戦値生成関数の出力は、 $n+1$ 個の1,0でない $q$ 以下の整数、基底生成関数の出力は、 $n+1$ 個の1,0でない $p$ 以下の整数で位数 $q$ の $F_p^*$ の元(位数 $p-1$ の乗法群の位数 $q$ 部分群の元)である整数である。

【0103】また、これらの関数は入出力間や出力の異なる成分間の関係を計算量的に意図して引数を決定できない関数とする。

【0104】基底生成関数の具体的な構成方法の例としては、 $|p|$ ビットを出力するハッシュ関数 $\text{Hash}(*)$ を一つ用意して、 $\text{Hash}(*)$

を計算し、次に、この計算結果をハッシュ関数の引数に入力してさらに計算結果を得る。これを、繰り返すことにより、再帰的に、数列 $h[0], h[1], h[2], \dots$ を生成し、

その各数値を $k$ 乗をした数列 $h[0]^k, h[1]^k, h[2]^k, \dots$ を求める。この中から順に、1,0でないものを $n+1$ 個選んでいく。

【0105】挑戦値生成関数の場合は、 $|q|$ ビットを出力するハッシュ関数を用いて数列を求め、その中から順に1,0でないものを $n+1$ 個選んでいく(この場合、 $k$ 乗する操作は必要無い)。

[公開鍵] 公開鍵について説明する。公開鍵は、二つの数値 $\eta[0, 0], \eta[0, 1]$ であり $\eta[0, 0]$ は位数 $q$ の $F_p^*$ の元とする。 $\eta[0, 1]$ は秘密鍵 $x$ を用いて、

$\eta[0, 1] = \eta[0, 0]^x / F_p^*$ と計算される。

【0106】[入力暗号文] 入力暗号文について説明する。平文を、 $p$ 以下で位数 $q$ の $F_p^*$ の元から選び、これを $M$ とする。これから疑似乱数生成器で生成した秘密乱数 $r$ を用いて、入力暗号文を、 $(\eta[0, 0]^r, M \eta[0, 1]^r) / F_p^*$ と計算する。

【0107】[再暗号化] 再暗号化について説明する。Elgamal暗号文 $(\eta[0, 0]^r, M \eta[0, 1]^r) / F_p^*$ が与えられた時、任意の乱数 $s$ を選んで、 $(\eta[0, 0]^r, M \eta[0, 1]^r) \rightarrow (\eta[0, 0]^r \eta[0, 0]^s, M \eta[0, 1]^r \eta[0, 1]^s) / F_p^* = (\eta[0, 0]^{r+s}, M \eta[0, 1]^{r+s}) / F_p^*$ なる変換を行うことを「再暗号化」という。上記変換は $r$ を知らなくても実行できる。またこの変換により再暗号化された暗号文の復号結果はかわらない。この時の乱数 $s$ を、「再暗号秘密乱数」と呼ぶ。

【0108】[置換行列] 置換行列について説明する。「置換行列」とはどの行にもまたどの列に対しても0でない成分が唯一存在し、1の値をとる。ただし本実施例では $F_q$ 上で考える。下に例をあげる。

【0109】

0 0 0 1 0  
1 0 0 0 0  
0 1 0 0 0  
0 0 0 0 1  
0 0 1 0 0 /  $F_q$

【0110】[準置換行列] 準置換行列について説明する。「準置換行列」とは、置換行列の1である成分を、 $F_p^*$ 上の3個ある1の3乗根のいずれかで置き換えたものと定義する。これらを $w, w^2, 1$ として下に準置換行列の例をあげる。

【0111】

0 0 0  $w^2$  0  
 $w$  0 0 0 0  
0  $w^2$  0 0 0  
0 0 0 0 1  
0 0  $w$  0 0 /  $F_q$

【0112】[再暗号シャッフル] 再暗号シャッフルについて説明する。入力暗号文列 $\eta[i, 0], \eta[i, 1]; i =$

$1, \dots, n$ の順序を入れ替えて、暗号文列  $\eta' [i, 0], \eta' [i, 1]; i=1, \dots, n$  を生成し、さらに  $n$  個の秘密乱数  $s[i]; i=1, \dots, n$  と、公開鍵  $\eta [0, 0], \eta [0, 1]$  を用いて、出力暗号文列  $g' [i, \Gamma]; i=1, \dots, n, \Gamma=0, 1$  を、  
 $g' [i, \Gamma] = \eta' [i, \Gamma] \eta [0, \Gamma]^{s[i]} / F_p^*$   $i=1, \dots, n, \Gamma=0, 1$

と計算する。これが、再暗号シャッフルの出力結果である。これを、「出力暗号文列」と呼ぶ。

【0113】〔再暗号シャッフル行列〕再暗号シャッフル行列について説明する。「再暗号シャッフル行列」とは、 $n+1$ 行 $n+1$ 列の行列で、その成分  $A[\mu, \nu]; \mu, \nu=0, \dots, n$  が、

$A[\mu, \nu] =$

$A[i, j] \quad i, j=1, \dots, n \quad \text{シャッフル行列307}$

$A[0, j] \in_{\mathcal{R}} \quad j=1, \dots, n \quad \text{再暗号秘密乱数305}$

$A[i, 0] \in_{\mathcal{R}} \quad i=1, \dots, n \quad \text{情報隠蔽因子306}$

$A[0, 0] \in_{\mathcal{R}} \quad \text{情報隠蔽因子306}$

であるものである。

【0114】〔再暗号シャッフル行列変換〕再暗号シャッフル行列変換について説明する。これは、入力文列  $g[\mu, \Gamma]$  に以下のように作用して、出力文列  $g'[\mu, \Gamma]$  を出力する。

【0115】 $g'[\mu, \Gamma] = \Pi_{\nu=0}^n g[\nu, \Gamma] A[\nu, \mu] / F_p^*$   $\mu=0, \dots, n, \Gamma=0, 1$

ここで、シャッフル行列が置換行列の場合、出力暗号文列を、 $g' [i, 0], g' [i, 1]; i=1, \dots, n$  とし、展開すると、ある置換  $(i, j) | \pi(i)=j$  に対して、

$g' [j, 0] = g[i, 0] \eta [0, 0]^{A[i, j]} / F_p^*$

$g' [j, 1] = g[i, 1] \eta [0, 1]^{A[i, j]} / F_p^*$

となり、これは再暗号シャッフルの出力となる。

【0116】またシャッフル行列が準置換行列の場合、準再暗号シャッフルの結果

$g' [j, 0] = g[i, 0]^{s[i]} \eta [0, 0]^{A[i, j]} / F_p^*$

$g' [j, 1] = g[i, 1]^{s[i]} \eta [0, 1]^{A[i, j]} / F_p^*$

を出力する(準再暗号シャッフルとは各出力暗号文を1または  $w$  または  $w^2$  乗すると再暗号シャッフルとなるものと定義する)。ここで  $w[i]; i=1, \dots, n$  は  $F_q$  上の1の三乗根のいずれかをとる。

【0117】〔実施例(1)〕本発明の一実施例をなす証明付再暗号シャッフル方法およびその検証方法について、図4、図5を参照して説明する。以下で、 $\Gamma=0, 1$  を取るものとする。

【0118】再暗号シャッフル情報401として、再暗号シャッフル行列402、係数基底404、元係数403を以下のように準備する。

【0119】再暗号シャッフル行列402に関しては、まず1から  $n$  までの数を順に並べる。疑似乱数発生器(不図示)を  $n$  回使って、 $n$  個数列を発生させ、その  $i$  番目の数を  $n-i+1$  で割り余りの数を求め  $\pi'(i)$  とする。

【0120】 $i$  は1から  $n$  迄順に、上記並べた数の下から

$\pi'(i)$  番目の数を  $\pi(i)$  とし、上記数列からこの数を取り除く作業を行い  $\pi(i); i=1, \dots, n$  を決定する。シャッフル行列の第  $i$  行目は  $\pi(i)$  列目の成分のみ値を1とし、その他を0とする。以上のようにして置換行列を生成する。

【0121】再暗号シャッフル行列のシャッフル行列以外の成分を以下のようにして生成する。まず、疑似乱数発生器により  $2n+1$  個の  $F_q$  上の数を作成し、 $A[i, 0], A[0, j], A[0, 0]; i, j=1, \dots, n$  に割り振る。以上を合わせて再暗号シャッフル行列とする。

【0122】係数基底404  $v$ 、元係数403  $r'[0]$  を生成に関しては、疑似乱数発生器で1, 0でない  $F_q$  上の数を生成し、 $r'[0]$  とし、疑似乱数発生器により  $F_p^*$  の元を生成し  $F_p^*$  上でその  $k$  乗をとり1, 0でないものを選び位数  $q$  の  $F_p^*$  の元を生成し、 $v$  とする。

【0123】 $r'[0] \in_{\mathcal{R}} F_q, \neq 0, 1$

$v \in_{\mathcal{R}} F_p^*, \neq 1, \text{ s.t. } v^q = 1 / F_p^*$

入力暗号文列  $\eta [i, 0], \eta [i, 1]; i=1, \dots, n$  と、公開鍵  $\eta [0, 0], \eta [0, 1]$  より、入力文列  $400g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$  を、

$g[0, \Gamma] = \eta [0, \Gamma] \quad \Gamma=0, 1$

$g[i, \Gamma] = \eta [i, \Gamma] / F_p^*, \quad i=1, \dots, n, \Gamma=0, 1$

とする。

【0124】以下、証明付再暗号シャッフル方法を使う。

【0125】変換情報保有コミットメント生成処理419における再暗号シャッフル行列作用405により、上記暗号シャッフル行列402を入力文列400に以下の様に作用させて、出力文列406  $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$  を、

$g'[\mu, \Gamma] = \Pi_{\nu=0}^n g[\nu, \Gamma]^{A[\nu, \mu]} / F_p^*, \quad \mu=0, \dots, n, \Gamma=0, 1$

と生成する。

【0126】ここで、 $g' [i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$  を出力暗号文列407、 $g' [0, \Gamma]; \Gamma=0, 1$  を変換情報保有コミットメント408とする。

【0127】変換条件コミットメント生成処理420における恒等式係数計算409により、元係数403  $r'[0]$  と、再暗号シャッフル行列402と、を用いて、恒等式係数410  $\phi[\mu], r'[0]$  を、

$r'[0] = r'[0]$

$\phi[0] = \sum_{j=1}^n A[j, 0] A[j, 0] + r'[0] A[0, 0] / F_q$

$\phi[i] = 2 \sum_{j=1}^n A[j, 0] A[j, i] + r'[0] A[0, i] / F_q, \quad i=1, \dots, n$

と生成する。

【0128】さらに、係数基底404  $v$  を用いて、隠蔽処理411により、恒等式係数410  $r'[0], \phi[0]$  を、

$v' = v^{r'[0]} / F_p^*$

$\omega = v^{\phi[0]} / F_p^*$

とコミットする。

【0129】以上より、 $\phi[i]$ ,  $\omega, v', v$  を変換条件コミットメント412とする。

【0130】ここで、コミットメント40 Aを、変換情報保有コミットメント408と変換条件コミットメント412とする。

【0131】応答生成処理421により、以上の入力文列400と、出力暗号文列417と、コミットメント409と、を挑戦値生成関数413の引数として、挑戦値414を、

$c[0]=1$ ,  
 $c[i]=\text{Hash}[i](g[v, 0], g[v, 1], g'[v, 0], g'[v, 1], v, \phi[v], \omega, v'; v=0, \dots, n) \quad i=1, \dots, n$   
 と生成し、この挑戦値から再暗号シャッフル行列02を用いて、応答416を  $r[\mu]=\sum_{v=0}^n A[\mu, v]c[v] / F_q \quad \mu=0, \dots, n$  と生成415する。

【0132】以上のコミットメント40 Aと応答416を、再暗号シャッフル証明文418として出力し、再暗号シャッフルの結果として出力暗号文列417を出力する。

【0133】検証方法について、図5を参照して説明する。

【0134】再暗号シャッフル検証方法により、入力文列400  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$  と、出力暗号文列417  $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$  と、再暗号シャッフル証明文418中のコミットメント409である変換情報保有コミットメント408  $g'[0, \Gamma]; \Gamma=0, 1$  と、変換条件コミットメント412  $\phi[v], \omega, v', v; v=0, \dots, n; \Gamma=0, 1$  と、を挑戦値生成関数500に代入して、挑戦値501を、

$c[0]=1$   
 $c[i]=\text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], \phi[v], \omega, v', v; v=0, \dots, n; \Gamma=0, 1) \quad i=1, \dots, n$

と生成する。

【0135】変換情報保有検証処理505により、この挑戦値501を用いて入力文列400と、変換情報保有コミットメント408と出力暗号文列417である出力文列406と応答416とを用いて検証式、

$\Pi_{\mu=0}^n g[\mu, \Gamma]^{r[\mu]} = \Pi_{\mu=0}^n g'[\mu, \Gamma]^{c[\mu]} / F_p$   
 $\Gamma=0, 1$

が成り立つことを確認502する。

【0136】変換条件検証処理506により、挑戦値501と応答416と変換条件コミットメント412とを用いて検証式  $v^{r[0]} v^{-\{\sum_{i=1}^n r[i]r[i]\}} = \omega v^{-\{\sum_{i=1}^n (c[i]c[i] + \phi[i]c[i])\}} / F_p$

が成り立つことを確認503する。

【0137】以上全ての検証式が成り立てば、証明文を受理504する。

【0138】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも正規直交行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。

【0139】入力暗号文と出力暗号文に制限が課せられており、この効果で、再暗号シャッフルの正当性を保証

できる場合には、本実施例により、証明付再暗号シャッフルを構成できる。

【0140】例えば、入力暗号文は限られた候補から選ばれていることが証明されていて、かつそれらの候補は互いに他を基底として表現できないとする。この入力暗号文を再暗号シャッフルしたのち、復号していずれの復号文も正しい候補から選ばれたものであったとき、本実施例による証明文からこの再暗号シャッフルが正当であることが言える。

【0141】なお図4における変換情報保有コミットメント処理419、変換条件コミットメント生成処理420、応答生成処理421は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図5における変換情報保有検証処理505、変換条件検証処理506は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピーディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0142】〔実施例(2)〕本発明の実施例(2)の証明付再暗号シャッフル方法およびその検証方法について、図6、図7を参照して説明する。以下では、 $\Gamma=0, 1$  を取るものとする。

【0143】再暗号シャッフル情報601として、再暗号シャッフル行列602、元係数603、係数基底604、605、準元係数606を、以下のように準備する。

【0144】まず再暗号シャッフル行列602に関しては、前記実施例(1)と同様に生成する。

【0145】元係数603  $\rho', \rho''$ 、係数基底604  $v$ 、係数基底605  $u$ 、準元係数606  $\lambda[\mu]; \mu=0, \dots, n$  に関しても、実施例(1)と同様な手法で、 $\rho', \rho'' \lambda[\mu]; \mu=0, \dots, n$  には1, 0でない  $F_q$  上の数を、係数基底  $u, v$  には位数  $q$  の  $F_p$  の元を生成する。

【0146】 $\rho' \in_r F_q, \neq 0, 1$

$\rho'' \in_r F_q, \neq 0, 1$

$v \in_r F_p, \neq 1, \text{ s.t. } v^q = 1 / F_p$

$\lambda[\mu] \in_r F_q, \neq 0, 1, \mu=0, \dots, n$

$u \in_r F_p, \neq 1, \text{ s.t. } u^q = 1 / F_p$

【0147】実施例(1)と同様にして、入力暗号文列と公開鍵より、入力文列600  $g[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$  を生成する。

【0148】以下、証明付再暗号シャッフル方法を用いる。

【0149】実施例(1)と同様に、変換情報保有コミットメント生成処理623を行い、出力文列603  $g'[\mu, \Gamma]; \mu=0, \dots, n; \Gamma=0, 1$  を、生成する。ここで、 $g'[i, \Gamma]; i=1, \dots, n; \Gamma=0, 1$  を、出力暗号文列604、 $g'[0,$



$\Gamma$ ];  $\Gamma=0, 1$ を交換情報保有コミットメント605とする。

【0150】変換条件コミットメント生成処理625における恒等式係数計算606により、元係数603  $\rho', \rho''$ 、と、再暗号シャッフル行列602を用いて恒等式係数607  $\psi[i], \phi[i], \phi[0], \rho', \rho''$ ;  $i=1, \dots, n$ を、  
 $\rho' = \rho'$   
 $\rho'' = \rho''$   
 $\psi[i] = \sum_{j=1}^n (3A[j, 0] + \rho'' \lambda[j])A[j, i] / F_q$   $i=1, \dots, n$   
 $\phi[i] = \sum_{j=1}^n (3A[j, 0]A[j, 0]A[j, i] + 2\rho'' \lambda[j]A[j, 0]A[j, i]) + \rho' A[0, i] / F_q$   $i=1, \dots, n$   
 $\phi[0] = \sum_{j=1}^n (A[j, 0]A[j, 0]A[j, 0] + \rho'' \lambda[j]A[j, 0]A[j, 0]) + \rho' A[0, 0] / F_q$   
と生成する。

【0151】さらに、係数基底604  $v$ を用いて、隠蔽処理608により、恒等式係数607  $\rho', \rho'', \phi[0]$ を、

$$\omega = v^{(0)} / F_p$$

$$v' = v^{(1)} / F_p$$

$$v'' = v^{(2)} / F_p$$

とコミット609する。さらに、係数基底605  $u$ を用いて、準元係数606  $\lambda[\mu]$ ;  $\mu=0, \dots, n$ を、

$$u[0] = u^{(0)} / F_p$$

$$u[i] = u^{(i)} / F_p \quad i=1, \dots, n$$

とコミット612する。

【0152】以上より、 $\psi[i], \phi[i], \omega, v', v'', v, u, u[0], u[i]$ ;  $i=1, \dots, n$ を変換条件コミットメント613とする。

【0153】ここで、コミットメント614を、交換情報保有コミットメント605と、変換条件コミットメント613とする。

【0154】応答生成処理624により、以上の入力文列600と、出力暗号文列604と、コミットメント614と、を挑戦値生成関数615の引数として、挑戦値616を、  
 $c[0]=1$

$$c[i] = \text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], u, u[v], v, \phi[j], \psi[j], \omega, v', v''; \Gamma=0, 1, 2; v=0, \dots, n; j=1, \dots, n) \quad i=1, \dots, n$$

と生成し、この挑戦値616から、再暗号シャッフル行列602を用いて、応答618を、

$$r[\mu] = \sum_{v=0}^n A[\mu, v]c[v] / F_q \quad \mu=0, \dots, n$$

と生成617する。

【0155】さらに、準応答620を、準元係数606  $\lambda[\mu]$ ;  $\mu=0, \dots, n$ と、応答618より、  
 $r' = \lambda[0] + \sum_{i=1}^n \lambda[i]r[i]r[i] / F_q$   
と生成619する。

【0156】以上のコミットメント614と、応答618と、準応答620と、を再暗号シャッフル証明文622として出力し、再暗号シャッフルの結果として、出力暗号文列604を出力する。検証方法について、図6及び図7を参照して、以下に説明する。

【0157】再暗号シャッフル検証方法により、入力文列600  $g[\mu, \Gamma]$ ;  $\mu=0, \dots, n$ ;  $\Gamma=0, 1$ と、出力暗号文列604  $g'[i, \Gamma]$ ;  $i=1, \dots, n$ ;  $\Gamma=0, 1$ と、再暗号シャッフル証明文622中のコミットメント614である交換情報保有コミットメント605  $g'[0, \Gamma]$ ;  $\Gamma=0, 1$ と、変換条件コミットメント609、912  $\psi[i], \phi[i], \omega, v', v'', v, u, u[0], u[i]$ ;  $i=1, \dots, n$ と、を挑戦値生成関数704に代入して挑戦値705を、

$$c[0]=1$$

$$c[i] = \text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], u, u[v], v, \phi[j], \psi[j], \omega, v', v''; \Gamma=0, 1, 2; v=0, \dots, n; j=1, \dots, n) \quad i=1, \dots, n$$

と生成する。

【0158】変換情報保有検証処理710により、この挑戦値705を用いて、入力文列600と、交換情報保有コミットメント605と、出力暗号文列604である出力文列603と、応答618と、を用いて検証式、

$$\prod_{\mu=0}^n g[\mu, \Gamma]^{r[\mu]} = \prod_{\mu=0}^n g'[\mu, \Gamma]^{c[\mu]} / F_p, \quad \Gamma=0, 1$$

が成り立つことを確認706する。

【0159】変換条件検証処理711により、挑戦値705と、応答618と、変換条件コミットメント609、612と、を用いて検証式、

$$v'^{(1)} v'^{(2)} v^{(0)} \{ \sum_{i=1}^n r[i]r[i]r[i] \} = \omega v^{(0)} \{ \sum_{i=1}^n (c[i]c[i]c[i] + \phi[i]c[i]c[i] + \phi[i]c[i]) \} / F_p$$

と検証式707

$$u^{(1)} = \prod_{i=1}^n u[i]^{r[i]} / F_p$$

が成り立つことを確認708する。

【0160】以上全ての検証式が成り立てば証明文を受理709する。

【0161】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。この時、出力暗号文列  $g'[i, \Gamma]$ ;  $i=1, \dots, n$ ;  $\Gamma=0, 1$ は、  
 $g'[j, 0] = g[i, 0]^{A[i, j]} g[0, 0]^{A[i, j]} / F_p$   
 $g'[j, 1] = g[i, 1]^{A[i, j]} g[0, 1]^{A[i, j]} / F_p$   
を出力した可能性を排除できない。ここで、 $w[i]$ が全て1の時が、再暗号シャッフルである。なお、 $w[i]$ ;  $i=1, \dots, n$ は、 $F_q$ 上の1の三乗根のいずれかをとる。

【0162】そこで、復号文として、 $F_q$ 上の1の三乗根乗の自由度を許すか、平文に決められた記号を記して三乗根乗の自由度を消せば、本実施例をもって、証明付再暗号シャッフルを構成できる。

【0163】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理623、変換条件コミットメント生成処理625、応答生成処理624は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理710、変換条件検証処理711は、コンピュータ



上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0164】〔実施例(3)〕本発明の実施例(3)として証明付再暗号シャッフル方法およびその検証方法について、図8、図9を参照して以下に説明する。以下で、 $\Gamma=0,1$ を取るものとする。また公開鍵は、 $\eta[-1, \Gamma]$ 、 $\eta[0, \Gamma]$ ； $\Gamma=0,1$ の二組あり、どちらも同じ秘密鍵を持つとする。

【0165】再暗号シャッフル情報801として、再暗号シャッフル行列802、元係数803、805、係数基底804、806、準元係数807を以下のように準備する。

【0166】本実施例で用いる再暗号シャッフル行列802は、実施例(1)と実施例(2)のものとは大きさが異なり、 $n+2$ 行 $n+1$ 列の行列である。

【0167】この再暗号シャッフル行列802を構成するシャッフル行列は、 $A[i, j]$ ； $i, j=1, \dots, n$ であり、再暗号秘密乱数は、 $A[-1, j]$ 、 $A[0, j]$ ； $j=1, \dots, n$ の $2 \times n$ 成分であり、知識の隠蔽因子は、 $A[\mu, 0]$ ； $\mu=-1, \dots, n$ の $n+2$ 成分である。これらの成分を、実施例(1)と同様に生成する。

【0168】元係数803  $r'[-1], r'[0]$ 、元係数805  $\rho, \rho', \rho''$ 、係数基底804  $v$ 、係数基底806  $u$ 、準元係数807  $\lambda[\mu]$ ； $\mu=0, \dots, n$ に関しても、実施例(1)と同様な手法で、 $r'[-1], r'[0], \rho, \rho', \rho'', \lambda[\mu]$ ； $\mu=0, \dots, n$ には、 $1, 0$ でない $F_q$ 上の数を、係数基底 $u, v$ には、位数 $q$ の $F_p$ の元を生成する。

【0169】 $r'[-1] \in_r F_q, \neq 0, 1$

$r'[0] \in_r F_q, \neq 0, 1$

$\rho \in_r F_q, \neq 0, 1$

$\rho' \in_r F_q, \neq 0, 1$

$\rho'' \in_r F_q, \neq 0, 1$

$v \in_r F_p, \neq 0, 1, \text{ s.t. } v^q = 1 / F_p$

$\lambda[\mu] \in_r F_q, \neq 0, 1 \mu=0, \dots, n$

$u \in_r F_p, \neq 0, 1, \text{ s.t. } u^q = 1 / F_p$

【0170】入力暗号文列  $\eta[i, 0], \eta[i, 1]$ ； $i=1, \dots, n$ と、公開鍵  $\eta[-1, \Gamma], \eta[0, \Gamma]$ ； $\Gamma=0, 1$ より、入力文列800  $g[\mu, \Gamma]$ ； $\mu=-1, \dots, n$ ； $\Gamma=0, 1$ を、  
 $g[-1, \Gamma] = \eta[-1, \Gamma] \quad \Gamma=0, 1$   
 $g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma] = \eta[i, \Gamma] / F_p \quad i=1, \dots, n, \Gamma=0, 1$ とする。

【0171】以下証明付再暗号シャッフル方法を用いる。

【0172】変換情報保有コミットメント生成処理832における再暗号シャッフル行列作用808により、上記再

暗号シャッフル行列802を入力文列800に以下の様に作用させて、出力文列809 $g'[\mu, \Gamma]$ ； $\mu=0, \dots, n$ ； $\Gamma=0, 1$ を、

$$g'[\mu, \Gamma] = \prod_{v=-1}^n g[v, \Gamma]^{A[\mu, v]} / F_p, \quad \mu=0, \dots, n, \Gamma=0, 1$$

と生成する。ここで、 $g'[i, \Gamma]$ ； $i=1, \dots, n$ ； $\Gamma=0, 1$ を出力暗号文列810、 $g'[0, \Gamma]$ ； $\Gamma=0, 1$ を変換情報保有コミットメント811とする。

【0173】変換条件コミットメント生成処理833、834における恒等式係数計算812、816により、元係数803、805  $r'[-1], r'[0], \rho, \rho', \rho''$ と、再暗号シャッフル行列802を用いて、恒等式係数817  $\psi[i], \phi[i], \phi[0], \rho, \rho', \rho''$ ； $i=1, \dots, n$ と、恒等式係数813  $\Phi[v], r'[0], r'[-1]$ ； $v=0, \dots, n$ を計算818、812する。

【0174】 $\rho = \rho$

$\rho' = \rho'$

$\rho'' = \rho''$

$$\psi[i] = \sum_{j=1}^n (3A[j, 0] + \rho'' \lambda[j]) A[j, i] / F_q, \quad i=1, \dots, n$$

$$\phi[i] = \sum_{j=1}^n (3A[j, 0] A[j, 0] A[j, i] + 2\rho'' \lambda[j] A[j, 0] A[j, i]) + \rho' A[0, i] + \rho A[-1, i] / F_q, \quad i=1, \dots, n$$

$$\phi[0] = \sum_{j=1}^n (A[j, 0] A[j, 0] A[j, 0] + \rho'' \lambda[j] A[j, 0] A[j, 0]) + \rho' A[0, 0] + \rho A[-1, 0] / F_q$$

$$r'[-1] = r'[-1]$$

$$r'[0] = r'[0]$$

$$\Phi[0] = \sum_{j=1}^n A[j, 0] A[j, 0] + r'[0] A[0, 0] + r'[-1] A[-1, 0] / F_q$$

$$\Phi[i] = 2 \sum_{j=1}^n A[j, 0] A[j, i] + r'[0] A[0, i] + r'[-1] A[-1, i] / F_q, \quad i=1, \dots, n$$

【0175】さらに係数基底804  $v$ を用いて、隠蔽処理814、818により、恒等式係数813、817  $r'[-1], r'[0], \Phi[0], \phi[0], \rho, \rho', \rho''$ を、

$$\omega = v^{(0)} / F_p$$

$$v' = v^{(1)} / F_p$$

$$v'' = v^{(2)} / F_p$$

$$\omega' = v^{(3)} / F_p$$

とコミット819し、

$$V = v^{(r'[-1])} / F_p$$

$$V' = v^{(r'[0])} / F_p$$

$$\Omega = v^{(\phi[0])} / F_p$$

とコミット815する。

【0176】さらに、係数基底806  $u$ を用いて、準元係数807  $\lambda[\mu]$ ； $\mu=0, \dots, n$ を、

$$u[0] = u^{(\lambda[0])} / F_p$$

$$u[i] = u^{(\lambda[i])} / F_p, \quad i=1, \dots, n$$

とコミット821、820する。

【0177】以上より、 $\Phi[i], V, V', \Omega, \psi[i], \phi[i], \omega, v', v'', \omega', v, u, u[0], u[i]$ ； $i=1, \dots, n$ を変換条件コミットメント822とする。

【0178】ここで、コミットメント823を、変換情報

保有コミットメント811と変換条件コミットメント822とする。

【0179】応答生成処理835により、以上の入力文列800と、出力暗号文列810と、コミットメント823を、挑戦値生成関数824の引数として、挑戦値825を、

$c[0]=1$   
 $c[i]=\text{Hash}[i](g[\mu, \Gamma], g'[\nu, \Gamma], u[\nu], u, \phi[j], \psi[j], \omega, \omega', v', v'', v, \Phi[j], \Omega, V', V; \mu=-1, \dots, n; \nu=0, \dots, n; j=1, \dots, n; \Gamma=0, 1, 2) i=1, \dots, n$   
 と生成し、この挑戦値825から、再暗号シャッフル行列802を用いて、応答827を、  
 $r[\mu]=\sum_{v=0}^n A[\mu, v]c[v] / F_q, \mu=-1, \dots, n$   
 と生成826する。

【0180】さらに、準応答829を、準元係数807  $\lambda[\mu]; \mu=0, \dots, n$ と応答827より、  
 $r'=\lambda[0]+\sum_{i=1}^n \lambda[i]r[i]r[i] / F_q$   
 と生成828する。

【0181】以上のコミットメント823と、応答827と、準応答829とを再暗号シャッフル証明文831として出力し、再暗号シャッフルの結果として、出力暗号文列810を出力する。

【0182】検証方法について図9を参照して説明する。

【0183】再暗号シャッフル検証方法により、入力文列800  $g[\mu, \Gamma]; \mu=-1, \dots, n; \Gamma=0, 1$ と、出力暗号文列810  $g'[\nu, \Gamma]; \nu=1, \dots, n; \Gamma=0, 1$ と、再暗号シャッフル証明文831中コミットメント823の変換情報保有コミットメント811  $g'[0, \Gamma]; \Gamma=0, 1$ と、変換条件コミットメント815、819、821  $\Phi[i], V', V, \Omega, \psi[i], \phi[i], \omega, v', v', \omega', v, u, u[0], u[i]; i=1, \dots, n$ と、を、挑戦値生成関数900に代入して、挑戦値901を、  
 $c[0]=1$

$c[i]=\text{Hash}[i](g[\mu, \Gamma], g'[\nu, \Gamma], u[\nu], u, \phi[j], \psi[j], \omega, \omega', v', v'', v, \Phi[j], \Omega, V', V; \mu=-1, \dots, n; \nu=0, \dots, n; j=1, \dots, n; \Gamma=0, 1, 2) i=1, \dots, n$   
 と生成する。

【0184】変換情報保有検証処理907により、この挑戦値901を用いて入力文列800と、変換情報保有コミットメント811と、出力暗号文列810である出力文列809と、応答827とを用いて検証式、

$$\prod_{\mu=-1}^n g[\mu, \Gamma]^{r[\mu]} = \prod_{\mu=0}^n g'[\mu, \Gamma]^{c[\mu]} / F_p, \Gamma=0, 1$$

が成り立つことを確認902する。

【0185】変換条件検証処理908、909により、挑戦値901と応答827と準応答829と変換条件コミットメント815、819、821とを用いて、検証式

$$v', r', v^{r[0]}, \omega^{r[1]}, v^{\sum_{i=1}^n r[i]r[i]} = \omega v^{\sum_{i=1}^n (c[i]c[i]c[i] + \psi[i]c[i]c[i] + \phi[i]c[i])} / F_p$$

が成り立つことを確認904し、検証式

$$u' = u[0] \prod_{i=1}^n u[i]^{r[i]r[i]} / F_p$$

が成り立つことを確認905し、検証式

$$v^{r[0]} v^{r[1]} v^{\sum_{i=1}^n r[i]r[i]} = \Omega v^{\sum_{i=1}^n (c[i]c[i] + \phi[i]c[i])} / F_p$$

が成り立つことを確認903する。

【0186】以上全ての検証式が成り立てば証明文を受理906する。

【0187】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【0188】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理832、変換条件コミットメント生成処理833、834、応答生成処理835は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理907、変換条件検証処理908、909は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0189】[実施例(4)] 本発明の実施例(4)の証明付再暗号シャッフル方法およびその検証方法について図10、図11を参照して以下に説明する。以下で、 $\Gamma=0, 1$ を取るものとする。また公開鍵は、 $\eta[0, \Gamma]; \Gamma=0, 1$ の一组である。

【0190】再暗号シャッフル情報1006として、再暗号シャッフル行列1001と、第2の情報隠蔽因子1004と、元係数1002、1005と、係数基底1003、1008と、準元係数1007を以下のように準備する。

【0191】まず再暗号シャッフル行列1001に関しては、前記実施例(1)と同様に生成し、これを $A[\mu, \nu]; \mu, \nu=0, \dots, n$ とする。

【0192】さらに、第2の情報隠蔽因子1004  $A[\nu, 0]; \nu=0, \dots, n$ を同様に生成する。

【0193】元係数1005  $\rho', \rho''$ 、元係数1002  $r'[0]$ 、係数基底1003  $v$ 、係数基底1008  $u$ 、準元係数1007  $\lambda[\mu]; \mu=0, \dots, n$  に関して、実施例(1)と同様な手法で、 $r'[0], \rho', \rho'', \lambda[\mu]; \mu=0, \dots, n$ には1, 0でない $F_q$ 上の数を、係数基底 $u, v$ には位数 $q$ の $F_p$ の元を生成する。

【0194】 $\rho' \in_r F_q, \neq 0, 1$

$\rho'' \in_r F_q, \neq 0, 1$

$r'[0], \in_r F_q, \neq 0, 1$

$v \in {}_R F_p^*$ ,  $\neq 1$ , s. t.  $v^q = 1 / F_p^*$   
 $\lambda[\mu] \in {}_R F_q$ ,  $\neq 0, 1$   $\mu = 0, \dots, n$   
 $u \in {}_R F_p^*$ ,  $\neq 1$ , s. t.  $u^q = 1 / F_p^*$

【0195】入力暗号文列  $\eta[i, 0]$ ,  $\eta[i, 1]$ ;  $i=1, \dots, n$ と、公開鍵  $\eta[0, \Gamma]$ ;  $\Gamma=0, 1$ より、入力文列1000  $g[\mu, \Gamma]$ ;  $\mu=0, \dots, n$ ;  $\Gamma=0, 1$ を、  
 $g[0, \Gamma] = \eta[0, \Gamma]$   $\Gamma=0, 1$   
 $g[i, \Gamma] = \eta[i, \Gamma]$   $i=1, \dots, n$ ,  $\Gamma=0, 1$   
 とする。

【0196】以下、証明付再暗号シャッフル方法を用いる。

【0197】変換情報保有コミットメント生成処理1042における再暗号シャッフル行列作用1009により、上記暗号シャッフル行列1001を入力文列1000に以下の様に作用させて、出力文列1010  $g'[\mu, \Gamma]$ ;  $\mu=0, \dots, n$ ;  $\Gamma=0, 1$ を、  
 $g'[\mu, \Gamma] = \Pi_{v=0}^{q-1} g[v, \Gamma]^{A[\mu, v, \Gamma]} / F_p^*$   $\mu=0, \dots, n$ ,  $\Gamma=0, 1$   
 と生成する。

【0198】ここで、 $g'[i, \Gamma]$ ;  $i=1, \dots, n$ ;  $\Gamma=0, 1$ 、出力暗号文列1011、 $g'[0, \Gamma]$ ;  $\Gamma=0, 1$ を第1の変換情報保有コミットメント1012とする。

【0199】第2の変換情報保有コミットメント生成処理1044により、さらに入力文列1000から選択1018して、第2の入力文列1019を $g[\mu, \Gamma']$ とする。ここでは、 $\Gamma'=0$ とする。

【0200】第2の変換情報保有コミットメント1021  $G'[0, \Gamma']$ を、  
 $G'[0, \Gamma'] = \Pi_{v=0}^{q-1} g[v, \Gamma']^{A[0, v, \Gamma']} / F_p^*$   $\Gamma'=0$ or1  
 と生成1020する。

【0201】変換条件コミットメント生成処理1045における恒等式係数計算1022により、元係数1005  $\rho'$ ,  $\rho''$ と、再暗号シャッフル行列1001と、を用いて、恒等式係数1023  $\psi[i]$ ,  $\phi[i]$ ,  $\phi[0]$ ,  $\rho'$ ,  $\rho''$ ;  $i=1, \dots, n$ を、  
 $\rho' = \rho'$   
 $\rho'' = \rho''$   
 $\psi[i] = \sum_{j=1}^n (3A[j, 0] + \rho'' \lambda[j]) A[j, i] / F_q$   $i=1, \dots, n$   
 $\phi[i] = \sum_{j=1}^n (3A[j, 0] A[j, 0] A[j, i] + 2\rho'' \lambda[j] A[j, 0] A[j, i]) + \rho' A[0, i] / F_q$   $i=1, \dots, n$   
 $\phi[0] = \sum_{j=1}^n (A[j, 0] A[j, 0] A[j, 0] + \rho'' \lambda[j] A[j, 0] A[j, 0]) + \rho' A[0, 0] / F_q$   
 と生成する。

【0202】さらに、係数基底1003vを用いて、隠蔽処理1024により、恒等式係数1023  $\phi[0]$ ,  $\rho'$ ,  $\rho''$ を、  
 $\omega = v^{q^{(0)}} / F_p^*$   
 $v' = v^{p^{(0)}} / F_p^*$   
 $v'' = v^{p^{(0)}} / F_p^*$   
 とコミット1025する。

【0203】さらに、係数基底1008uを用いて、準元係

数1007  $\lambda[\mu]$ ;  $\mu=0, \dots, n$ を、  
 $u[0] = u^{q^{(0)}} / F_p^*$   
 $u[i] = u^{q^{(i)}} / F_p^*$   $i=1, \dots, n$   
 とコミット1027する。

【0204】変換条件コミットメント生成処理1043における恒等式係数計算1013により、元係数1002  $r'[0]$ と再暗号シャッフル行列1001と第2の情報隠蔽因子1004とを用いて恒等式係数1014  $\Phi[v]$ ,  $r'[0]$ ;  $v=0, \dots, n$ を、  
 $r'[0] = r'[0]$   
 $\Phi[0] = \sum_{j=1}^n B[j, 0] B[j, 0] + r'[0] B[0, 0] / F_q$   
 $\Phi[i] = 2 \sum_{j=1}^n B[j, 0] A[j, i] + r'[0] A[0, i] / F_q$   $i=1, \dots, n$   
 と生成する。

【0205】さらに係数基底1003 vを用いて、隠蔽処理1015により、恒等式係数1014  $r'[0]$ ,  $\Phi[0]$ を、  
 $V' = v^{r' \cdot (0)} / F_p^*$   
 $\Omega = v^{\Phi \cdot (0)} / F_p^*$   
 とコミット1016する。

【0206】以上により、第1の変換条件コミットメント1028を、 $\psi[i]$ ,  $\phi[i]$ ,  $\omega$ ,  $v'$ ,  $v'$ ,  $v$ ,  $u$ ,  $u[0]$ ,  $u[i]$ ;  $i=1, \dots, n$ とする。第2の変換条件コミットメント1016を、 $\Phi[i]$ ,  $V'$ ,  $\Omega$ ,  $v$ ;  $i=1, \dots, n$ とする。

【0207】ここで、第1のコミットメント1017を、第1の変換情報保有コミットメント1012と第1の変換条件コミットメント1028とし、第2のコミットメント1029を第2の変換情報保有コミットメント1021と第2の変換条件コミットメント1016とする。

【0208】応答生成処理1046により、以上の入力文列1000と、出力暗号文列1011と、第1コミットメント1017とを、挑戦値生成関数1030の引数として、第1の挑戦値1031を、  
 $c[0] = 1$

$c[i] = \text{Hash}[i](g[v, \Gamma], g'[v, \Gamma], u[v], u, \phi[j], \psi[j], \omega, v', v', v; v=0, \dots, n; j=1, \dots, n; \Gamma=0, 1)$   $i=1, \dots, n$   
 と生成し、この挑戦値1031から、再暗号シャッフル行列1001を用いて、第1の応答1033を、  
 $r[\mu] = \sum_{v=0}^{q-1} A[\mu, v] c[v] / F_q$   $\mu=0, \dots, n$   
 と生成1032する。

【0209】さらに、準応答1039を、準元係数1007  $\lambda[\mu]$ ;  $\mu=0, \dots, n$ と、応答1033より、  
 $r' = \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] / F_q$   
 と生成1038する。

【0210】応答生成処理1047により、第2の入力文列1019と、出力暗号文列1011と、第2コミットメント1029とを、挑戦値生成関数1034の引数として、第2の挑戦値1035を、  
 $C[0] = 1$   
 $C[i] = \text{Hash}[i](g[v, \Gamma'], G'[0, \Gamma'], g'[j, \Gamma'], \Phi[j], \Omega, V'; v=0, \dots, n; j=1, \dots, n; \Gamma'=0)$   $i=1, \dots, n$

と生成し、この挑戦値1035から、再暗号シャッフル行列1001と、第2の情報隠蔽因子1004を用いて、第2の応答1037を、

$$R[\mu] = B[\mu, 0] + \sum_{i=1}^n A[\mu, i] C[i] / F_p \quad \mu = 0, \dots, n$$

と生成1036する。

【0211】以上のコミットメント1017、1029と、応答1033、1037と、準応答1039と、を、再暗号シャッフル証明文1040として出力し、再暗号シャッフルの結果として出力暗号文列1011を出力する。

【0212】検証方法について、図11を参照して説明する。

【0213】再暗号シャッフル検証方法により、入力文列1000と、出力暗号文列1011と、再暗号シャッフル証明文1040の第1のコミットメント1012、1025、1027を、挑戦値生成関数1100に代入して、第1の挑戦値1101を、 $C[0]=1$

$C[i] = \text{Hash}[i]$  (入力文列、出力暗号文列、第1のコミットメント)  $i=1, \dots, n$  と生成する。

【0214】さらに、第2の入力文列1019と、再暗号シャッフル証明文1040の第2のコミットメント1016、1021と、出力暗号文列1011とを、挑戦値生成関数1108に代入して第2の挑戦値1109を

$$c[0]=1$$

$c[i] = \text{Hash}[i]$  (第2の入力文列、出力暗号文列、第2のコミットメント)  $i=1, \dots, n$

と生成する。

【0215】変換情報保有検証処理1112により、第1の挑戦値1101を用いて、入力文列1000と、第1の変換情報保有コミットメント1012と、出力暗号文列1011と、第1の応答1033と、を用いて検証式、

$$\Pi_{\mu=0}^n g[\mu, \Gamma]^{\Gamma} = \Pi_{\mu=0}^n g'[\mu, \Gamma]^{\Gamma} / F_p$$

$$\Gamma = 0, 1$$

が成り立つことを確認1103する。

【0216】変換情報保有検証処理1113により、第2の挑戦値1109を用いて第2の入力文列1019と、第2の変換情報保有コミットメント1021と、出力暗号文列1011と、第2の応答1037と、を用いて第2の知識検証式、

$$\Pi_{\mu=0}^n g[\mu, \Gamma']^{\Gamma'} = G'[0, \Gamma'] \Pi_{i=1}^n g'[i, \Gamma']^{c[i]} / F_p \quad \Gamma' = 0$$

が成り立つことを確認1105する。

【0217】変換条件検証処理1111により、第1の挑戦値1101と、第1の応答1033と、第1の変換条件コミットメント1025とを用いて、検証式1102、

$$v^{r[0]} \cdot v^{\sum_{i=1}^n r[i]r[i]} = \omega v^{\sum_{i=1}^n (c[i]c[i]c[i] + \phi[i]c[i]c[i] + \phi[i]c[i])} / F_p$$

と、準応答1039と、準応答コミットメント1027と、第1の応答1033と、検証式1107、

$$u^{r'} = u[0] \Pi_{i=1}^n u[i]^{r[i]r[i]} / F_p$$

が成り立つことを確認する。

【0218】変換条件検証処理1114により、第2の挑戦値1109と、第2の応答1037と、第2の変換条件コミットメント1016とを用いて、検証式1106、

$$v^{r[0]} \cdot v^{\sum_{i=1}^n R[i]R[i]} = \Omega v^{\sum_{i=1}^n (C[i]C[i] + \Phi[i]C[i])} / F_p$$

が成り立つことを確認する。

【0219】以上全ての検証式が成り立てば証明文を受理1110する。

【0220】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【0221】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理1042、変換条件コミットメント生成処理1043、1045、応答生成処理1046、1047は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理1112、1113、変換条件検証処理1111、1114は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0222】[実施例(5)] 本発明の実施例(5)をなす入力文列生成方法について、図12を参照して、以下に説明する。 $\Gamma$ は、0、1、2の値をとる。

【0223】公開鍵302  $g[0, 0]$ 、 $g[0, 1]$ に対応する秘密鍵 $x$ は、 $t$ 人の証明者により分散所持されている。

【0224】この時、秘密鍵 $x[\Lambda]$  ( $\Lambda=1, \dots, t$ )により、各証明者の公開鍵を、 $g[0, 0]$ 、 $g[0, 1, \Lambda] = g[0, 0]^{x[\Lambda]}$  ( $\Lambda=1, \dots, t$ )とし、全体の公開鍵は、 $g[0, 0]$ 、 $g[0, 1] = \Pi_{\Lambda=1}^t g[0, 1, \Lambda]$ とする。

【0225】入力暗号文列301  $\eta[i, 0]$ 、 $\eta[i, 1]$  ( $i=1, \dots, n$ )と、公開鍵302  $\eta[0, 0]$ 、 $\eta[0, 1]$ が入力され、公開鍵302およびElgamal領域変数 $p$ 、 $q$ より、基底生成関数1200で、入力ベクトル1201を生成し、入力文列300  $g[\mu, \Gamma]$  ( $\mu=0, \dots, n$ ;  $\Gamma=0, 1, 2$ )を、

$$g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$$

$$g[i, \Gamma] = \eta[i, \Gamma] / F_p \quad i=1, \dots, n, \quad \Gamma=0, 1$$

$$g[\mu, 2] = \text{Hash}[\mu](p, q, \eta[0, 0], g[0, 1, \Lambda]; \Lambda=1, \dots, t) \quad \mu=0, \dots, n$$

とする。

【0226】前記実施例(1)から実施例(4)に、本実施例の入力文列生成方法を適用する場合、これらの実施例における $\Gamma$ の値をとる範囲を、全て0、1から、0、1、2に

変更する。この新たに導入された、入力暗号文でも公開鍵でもない $\Gamma=2$ の成分が、入力暗号文生成者にも意図できない入力文列の成分となり、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【0227】また前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力文列を、 $g[-1, \Gamma]$ まで拡張して、公開鍵  $g[-1, 0]$ 、 $g[-1, 1]$ 、 $\eta[0, 0]$ 、 $\eta[0, 1]$ より、  
 $g[-1, \Gamma] = \eta[-1, \Gamma] \quad \Gamma=0, 1$   
 $g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma] = \eta[i, \Gamma] / F^*$   $i=1, \dots, n, \Gamma=0, 1$   
 $g[\mu, 2] = \text{Hash}[\mu](p, q, \eta[0, 0], g[0, 1, \Lambda]; \Lambda=1, \dots, t)$   $\mu=1, \dots, n$   
 とする。

【0228】また実施例(4)に、本実施例の入力文列生成方法を適用する場合、 $\Gamma'=2$ とし、第2の情報隠蔽因子より、第2の変換情報保有コミットメントを、  
 $G'[02] = \Pi_{v=1}^n g[v, 2]^{A^{(v, 0)}} / F^*$   
 $G'[i2] = \Pi_{v=1}^n g[v, 2]^{A^{(v, i)}}$   $i=1, \dots, n$   
 と変更する。

【0229】さらに、証明付再暗号シャッフル方法または再暗号シャッフル検証方法において、第2の入力文列  $g[\mu, \Gamma']$ ； $\Gamma=2$ と、第2のコミットメントとを挑戦値生成関数の引数として、第2の挑戦値を、  
 $C[0]=1$   
 $C[i] = \text{Hash}[i](g[v, 2], G'[v, 2], \Phi[j], \Omega, V'; v=0, \dots, n; j=1, \dots, n; \Gamma=0, 1)$   $i=1, \dots, n$   
 と変更する。

【0230】さらに、変換情報保有検証処理における第2の知識検証式は、  
 $\Pi_{\mu=0}^n g[\mu, 2]^{C^{(\mu, \Gamma')}} = \Pi_{\mu=0}^n G'[\mu, \Gamma']^{C^{(\mu, \Gamma')}} / F^*$   
 と変更する。

【0231】〔実施例(6)〕本発明の実施例(6)をなす入力文列生成方法について、図13と図14を参照して説明する。 $\Gamma$ は、0、1の値をとる。

【0232】前記実施例(5)と同様に、秘密鍵 $x$ は、 $t$ 人の証明者により分散所持されている。各証明者 $\Lambda$ ； $\Lambda=1, \dots, t$ は、証明付公開鍵列方法1304により、入力暗号文列301  $\eta[i, 0]$ 、 $\eta[i, 1]$ ； $i=1, \dots, n$ と、公開鍵302  $\eta[0, 0]$ 、 $\eta[0, 1]$ とを、共通初期値1310として、秘密鍵1301 $x[\Lambda]$ と、疑似秘密鍵1302 $\alpha[\Lambda]$ と、を公開鍵列情報1300として入力し、分散公開鍵列対1305  $g'[\mu, 1, \Lambda]$ ； $\mu=0, \dots, n$ と、公開鍵列証明文1306と、を得る。

【0233】公開鍵列検証方法1307により、各証明者の出力した分散公開鍵列対1305と、公開鍵列証明文と、共通初期値1310とから、分散公開鍵列1305の正当性が検証されたら、前処理方法により、各証明者の分散公開鍵列対1305  $g'[\mu, 1, \Lambda]$ ； $\mu=0, \dots, n$ ； $\Lambda=1, \dots, t$ を合わせ

て、公開鍵列対140  $3g'[\mu, 1, \Lambda]$ ； $\mu=0, \dots, n$ を、  
 $g'[\mu, 1] = \Pi_{\Lambda=1}^t g'[\mu, 1, \Lambda] / F^*$   $\mu=0, \dots, n$   
 とする。ここで、 $g'[0, 1] = \eta[0, 1]$ に入れ替える。

【0234】共通初期値である入力暗号文列301  $\eta[i, 0]$ 、 $\eta[i, 1]$ ； $i=1, \dots, n$ と、公開鍵302  $\eta[0, 0]$ 、 $\eta[0, 1]$ とから、公開鍵列底1401  $g'[\mu, 0]$ ； $\mu=0, \dots, n$ を、  
 $g'[0, 0] = \eta[0, 0]$   
 $g'[i, 0] = \text{Hash}[i](\eta[0, 0], \eta[0, 1, \Lambda], \eta[j, \Gamma]; \Lambda=1, \dots, t; \Gamma=0, 1; j=1, \dots, n)$   $i=1, \dots, n$   
 と生成1400する。ここでも、公開鍵列対1403と同様に $g'[0, 0]$ が入れ替えている。

【0235】公開鍵列底1401と公開鍵列対1403を合わせて、公開鍵列1404  $g'[\mu, \Gamma]$ ； $\mu=0, \dots, n, \Gamma=0, 1$ とする。

【0236】公開鍵列1404と、入力暗号文列301と、公開鍵302とから、入力文列300  $g[\mu, \Gamma]$ ； $\mu=0, \dots, n$ ； $\Gamma=0, 1$ を、  
 $g[0, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma] = \eta[i, \Gamma] g'[i, \Gamma] / F^*$   $i=1, \dots, n, \Gamma=0, 1$   
 とする（前処理1402）。

【0237】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力暗号文列  $\eta[i, \Gamma]$ ； $i=1, \dots, n$ ； $\Gamma=0, 1$ と、公開鍵  $\eta[0, \Gamma]$ ； $\Gamma=0, 1$ に対して、公開鍵列  $g'[\mu, \Gamma]$ ； $\mu=1, \dots, n$ ； $\Gamma=0, 1$ を生成する。ただし、 $g'[0, \Gamma]$ ； $\Gamma=0, 1$ は、公開鍵に等しい。そして、入力文列  $g[\mu, \Gamma]$ ； $\mu=1, \dots, n$ ； $\Gamma=0, 1$ を、  
 $g[-1, \Gamma] = \eta[0, \Gamma] \quad \Gamma=0, 1$   
 $g[i, \Gamma] = \eta[i, \Gamma] g'[i, \Gamma] / F^*$   $i=0, \dots, n, \Gamma=0, 1$   
 とする。

【0238】本実施例では、新たに生成された公開鍵列が、入力暗号文生成者にも意図できないため、入力暗号文にそれを乗じた入力文列の成分も意図できない。そのため、証明者が生成できる応答に制限を課す働きをなし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【0239】なお図13に示した、証明付公開鍵列装置1304、前処理装置1309、公開鍵列検証装置1307は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0240】〔実施例(7)〕本発明の実施例(7)の入力文列生成方法について、図15乃至図18を参照して説明する。 $\Gamma$ は、0、1の値をとる。前記実施例(5)と同様に、秘密鍵1502  $x$ は、 $t$ 人の証明者により、分散所持されている。

【0241】各証明者 $\Lambda$ ; $\Lambda=1, \dots, t$ は、証明付公開鍵列方法1504により、Elgamal領域変数を共通初期値1500として、秘密鍵1502  $x[\Lambda]$ と、疑似秘密鍵1503  $\alpha[\Lambda]$ と、を、公開鍵列情報1501として入力し、分散公開鍵列対1505  $g'[\mu, 1, \Lambda]$ ;  $\mu=0, \dots, n$ と、公開鍵列証明文1506と、を得る。

【0242】公開鍵列検証方法1507により、各証明者の出力した分散公開鍵列対1505と、公開鍵列証明文1506と、共通初期値1500より、分散公開鍵列対1505の正当性が検証1508されたら、各証明者の分散公開鍵列対1505  $g'[\mu, 1, \Lambda]$ ;  $\mu=0, \dots, n$ ;  $\Lambda=1, \dots, t$ を合わせて、公開鍵列対1509  $g'[\mu, 1, \Lambda]$ ;  $\mu=0, \dots, n$ を、 $g'[\mu, 1] = \prod_{\Lambda=1}^t g'[\mu, 1, \Lambda] / F_p$ ,  $\mu=0, \dots, n$ とする。

【0243】共通初期値1500から公開鍵列底 $g'[\mu, 0]$ ;  $\mu=0, \dots, n$ を、 $g'[\mu, 0] = \text{Hash}'[\mu](p, q)$   $\mu=0, \dots, n$ と生成する。

【0244】公開鍵列底と公開鍵列対1509を合わせて、公開鍵列1611  $g'[\mu, \Gamma]$ ;  $\mu=0, \dots, n, \Gamma=0, 1$ とする。

【0245】各入力暗号文生成者 $i=1, \dots, n$ は、証明付暗号化方法1606により、平文1602 $m[i]$ と、個別公開鍵1601 $g'[i, \Gamma]$ ;  $\Gamma=0, 1$ と、秘密乱数1604 $s[i]$ と、疑似秘密乱数1605 $s'[i]$ とより、入力暗号文1607  $\eta[i, \Gamma]$ ;  $\Gamma=0, 1$ を、 $\eta[i, 0] = g'[i, 0]^{s[i]} / F_p$ ,  $\eta[i, 1] = m[i] g'[i, 1]^{s[i]} / F_p$ と生成する。

【0246】またコミットメント(疑似暗号文底1704)、挑戦値1707、応答1709を順に以下のように生成し、 $\eta[i2] = g'[i, 0]^{s'[i]} / F_p$ ,  $c'[i] = \text{Hash}[0](\eta[i, 0], \eta[i, 1], \eta[i2])$ ,  $\theta'[i] = c'[i] s[i] + s'[i] / F_q$ 疑似暗号文底1704と応答1709とを暗号化証明文1608とする。

【0247】暗号化検証装置により、全ての入力暗号文1607と暗号化証明文1608に関して、 $c'[i] = \text{Hash}[0](\eta[i, 0], \eta[i, 1], \eta[i2])$ と、挑戦値1801を求め、これに応答1709を用いて、検証式1802

$\eta[i, 0]^{c'[i]} = \eta[i, 1]^{c'[i]} \eta[i2] / F_p$ ,  
が成り立つことを確認1610する。入力暗号文1607全ての正当性が確認されたら、入力暗号文323  $\eta[i, \Gamma]$ ;  $\Gamma=0, 1$ と、共有公開鍵1600  $g'[0, \Gamma]$ ;  $\Gamma=0, 1$ とより入力文列300を、 $g[0, \Gamma] = g'[0, \Gamma]$   
 $g[i, \Gamma] = \eta[i, \Gamma]$   $i=1, \dots, n$ とする。

【0248】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合は、

$g[-1, \Gamma] = g'[-1, \Gamma]$   
 $g[0, \Gamma] = g'[0, \Gamma]$   
 $g[i, \Gamma] = \eta[i, \Gamma]$   $i=1, \dots, n$   
とする。

【0249】本実施例では、始めに生成された公開鍵列が、入力暗号文生成者にも意図できないため、これを基に暗号化したことが主命されている入力暗号文の成分も、意図できない。そのため、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【0250】なお図15に示した、証明付公開鍵列装置1504、公開鍵列検証装置1507は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図16乃至図18に示した、証明付暗号化装置1606、暗号化検証装置1609は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体(例えばCD-ROM、DVD(digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか)から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0251】[実施例(8)] 本発明の実施例(8)における証明付公開鍵列方法について図19と図20とを参照して説明する。

【0252】共通初期値 $e$ と、秘密鍵1902  $x$ と、疑似秘密鍵1903  $\alpha$ とが、公開鍵列情報1901として、入力される。

【0253】共通初期値1900から、公開鍵列底1905  $g'[\mu, 0]$ ;  $\mu=0, \dots, n$ を、 $g'[\mu, 0] = \text{Hash}'[\mu](e)$   $\mu=0, \dots, n$ と生成1904する。

【0254】これから、秘密鍵1902  $x$ と、疑似秘密鍵1903  $\alpha$ とにより、(分散)公開鍵列対1907  $g'[\mu, 1]$ ;  $\mu=0, \dots, n$ が、 $g'[\mu, 1] = g'[\mu, 0]^x / F_p$ ,  $\mu=0, \dots, n$ と生成1906され、疑似公開鍵列対1909が、 $g'[\mu, 2] = g'[\mu, 0]^\alpha / F_p$ ,  $\mu=0, \dots, n$ と生成1908される。

【0255】挑戦値1912、応答1914を順に、 $c'' = \text{Hash}[0](g'[\mu, 0], g'[\mu, 2])$ ;  $\mu=0, \dots, n$   
 $\theta = c'' x + \alpha / F_q$   
と生成し、疑似公開鍵列1909と応答1914を公開鍵列証明文1915とする。

【0256】公開鍵列検証方法により、挑戦値2003を、 $c'' = \text{Hash}[0](g'[\mu, 0], g'[\mu, 2])$ ;  $\mu=0, \dots, n$ と生成2000し、応答1914を用いて検証式、 $g'[\mu, 0]^\theta = g'[\mu, 0]^{c''} g'[\mu, 2] / F_p$ ,  $\mu=0, \dots, n$ を検証2004する。

【0257】本実施例では、始めに生成された公開鍵列底、誰にも意図できないため、これを元に作られた公開鍵列の成分も意図できない。

【0258】なお図19、図20に示した、証明付公開鍵列装置、公開鍵列検証装置は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばCD-ROM、DVD (digital versatile disk)、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【0259】〔実施例(9)〕本発明の実施例(9)として、証明付復号について説明する。前記実施例(5)と同様に、秘密鍵  $x$  は、 $t$  人の証明者により分散所持されている。

【0260】 $\Lambda; \Lambda=1, \dots, t$  番目の証明者は、 $\Lambda-1$  番目の証明者による部分復号の結果を入力して、それを部分復号する。 $\Lambda$  番目の証明者による部分復号の結果が復号文である。ただし、0番目の証明者による部分復号の結果は、上記最終的な再暗号シャッフルの出力のことである。

【0261】ここで、 $\Lambda$  番目の証明者の行う証明付部分復号(部分復号およびその証明文の提出)について説明する。

【0262】疑似乱数発生器により1,0でない $F_q$ 上の数を $\beta[\Lambda]$ を作成する。

【0263】 $\beta[\Lambda] \in_r F_q, \neq 0, 1$

【0264】また、自身の公開鍵  $g[0, 0], g'[0, 1, \Lambda]$  を、 $g[0, 0], g[0, 1]$  と、入力された暗号文列を、 $g[i, \Gamma]; i=1, \dots, n, \Gamma=0, 1$  とし、自身の公開鍵と秘密鍵 $x[\Lambda]$ から、部分復号基底 $G[\mu, 0, \Lambda]; \mu=0, \dots, n$  と、疑似部分復号基底 $G[\mu, 1, \Lambda]; \mu=0, \dots, n$  を、

$$G[\mu, 0, \Lambda] = g[\mu, 0]^{r[\mu, \Lambda]} / F_p^* \quad \mu=0, \dots, n \quad *$$

$$\prod_{\mu=1}^{n+1} g[\mu, \Gamma]^{r[\mu, \Gamma]} = \prod_{\mu=1}^{n+1} g[\mu, 0]^{r[\mu, \Gamma]} / F_p^*$$

$$= \prod_{v=1}^{n+1} (\prod_{\mu=1}^{n+1} g[\mu, \Gamma]^{A[\mu, v]})^{c[v]} / F_p^*$$

$$= \prod_{v=1}^{n+1} g'[v, \Gamma]^{c[v]} / F_p^*$$

より分かる。

【0272】準元係数をコミットしたものと、これに付随する応答と準応答が検証式を満たすことは、

$$u^{r[\cdot]} = u^{\wedge} \{ \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] \} / F_p^*$$

$$= u^{\lambda[0]} \prod_{i=1}^n (u^{\lambda[i]})^{r[i] r[i]} / F_p^*$$

$$= u[0] \prod_{i=1}^n u[i]^{r[i] r[i]} / F_p^* \quad *$$

$$v^{r[\cdot]} = \prod_{i=1}^n v^{r[i] r[i]} / F_p^*$$

$$= (v^{r[0]})^{r[0]} v^{\wedge} \{ \sum_{i=1}^n \sum_{\mu=0}^n \sum_{v=0}^n A[i, \mu] A[i, v] c[\mu] c[v] \} / F_p^*$$

$$= v^{\wedge} \{ r[0] \sum_{\mu=0}^n [0, \mu] c[\mu] + 2 \sum_{i=1}^n \sum_{j=1}^n A[i, 0] A[i, j] c[j] + \sum_{i=1}^n A[i, 0] A[i, 0] + \sum_{i=1}^n \sum_{j=1}^n A[i, j] A[i, k] c[j] c[k] \} / F_p^*$$

$$= v^{\wedge} \{ \sum_{i=1}^n \phi[i] c[i] + \phi[0] + \sum_{i=1}^n c[i] c[i] \} / F_p^*$$

$$= \omega v^{\wedge} \{ \sum_{i=1}^n (c[i] c[i] + \phi[i] c[i]) \} / F_p^*$$

\*  $G[\mu, 1, \Lambda] = g[\mu, 0]^{\beta[\Lambda]} / F_p^* \quad \mu=0, \dots, n$   
と生成する。コミットメントとして、 $g[\mu, \Gamma, \Lambda]; \mu=0, \dots, n, \Gamma=0, 1, \Lambda=0, \dots, t$  を出力する。

【0265】 $g[0, 1, \Lambda] = g[0, 0]^{\beta[\Lambda]} = G[0, 0, \Lambda]$  は、公開鍵と重複しているが同じものが計算されている。

【0266】挑戦値を、

$c[\Lambda] = \text{Hash}[0](g[\mu, 0], G[\mu, \Gamma, \Lambda]; \mu=0, \dots, n; \Gamma=0, 1)$

と生成し、これを用いて、応答 $r[\Lambda]$ を

$$r[\Lambda] = \beta[\Lambda] + c[\Lambda] x[\Lambda] / F_q$$

と生成して出力する。部分復号基底、疑似部分復号基底と応答を証明付部分復号の証明文として出力する。

【0267】部分復号を

$$g[i, 0] \rightarrow g[i, 0] \quad i=1, \dots, n$$

$$g[i, 1] \rightarrow g[i, 1] / G[i, 0, \Lambda] / F_p^* \quad i=1, \dots, n$$

として出力する。

【0268】検証処理は、入力暗号文列と証明文中とより挑戦値を、

$$c[\Lambda] = \text{Hash}[0](g[\mu, 0], G[\mu, \Gamma, \Lambda]; \mu=0, \dots, n; \Gamma=0, 1)$$

と生成し、証明文中の応答、入力暗号文列、分復号基底、疑似部分復号基底を用いて、

$$g[\mu, 0]^{r[\mu, \Lambda]} = G[\mu, 0, \Lambda]^{r[\mu, \Lambda]} G[\mu, 1, \Lambda] / F_p^* \quad \mu=0, \dots, n$$

を確認し、さらに部分復号が、この $G[\mu, 0, \Lambda]$ を用いて行われたことを確認して受理する。

【0269】以上を、 $t$  人の証明者全てにより行われた結果を復号文とする。

【0270】〔正当性〕以上説明した実施例の正当性について説明する。

【0271】〔完全性〕入力文列と、出力暗号文列と変換情報保有コミットメントとである出力文列と、これに付随する応答と挑戦値が変換情報保有検証処理の検証式を満たすことは、

$$g[\mu, \Gamma]^{\wedge} \{ \sum_{v=1}^{n+1} A[\mu, v] c[v] \}$$

※よりわかる。

【0273】変換条件コミットメント生成処理が出力した恒等式の係数と、これに付随する応答と挑戦値が知識検証処理の検証式を満たすことは、以下のようにして分かる。

【0274】実施例(1)の恒等式係数に関しては、



より成立することがわかる。以上、 $A[i, j]$ が置換行列であるという事実を使った。

【0275】前記実施例(2)の恒等式係数に関しては、

$$\begin{aligned} & \sum_{i=1}^n r[i]r[i]r[i] + \sum_{i=1}^n \rho'' \lambda[i]r[i]r[i] + \rho' r[0] / F_p \\ &= \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[h, i]A[h, j]A[h, k]c[i]c[j]c[k] \\ &+ \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n (3A[h, 0]A[h, i]A[h, j] + \rho'' \lambda[h]A[h, i]A[h, j])c[i]c[j] \\ &+ \sum_{h=1}^n \sum_{i=1}^n (3A[h, 0]A[h, 0]A[h, i] + 2\rho'' \lambda[h]A[h, 0]A[h, i] + \rho' A[0, i])c[i] \\ &+ \sum_{h=1}^n (A[h, 0]A[h, 0]A[h, 0] + \rho'' \lambda[h]A[h, 0]A[h, 0] + \rho' \lambda[0] + \rho' A[0, 0]) \\ &/ F_q \\ &= \sum_{h=1}^n (c[h]c[h]c[h] + \psi[h]c[h]c[h] + \phi[i]c[i] + \phi[0]) / F_q \end{aligned}$$

であり、これは、 $v \in \{ \sum_{h=1}^n (c[h]c[h]c[h] + \psi[h]c[h]c[h] + \phi[i]c[i]) \} \omega[0] / F_p$ の指数部と等しい。

【0276】以上で最後の式を導くために $A[i, j]$ が置換行列であるという事実を使った。

【0277】前記実施例(3)、及び実施例(4)に関しても同様の議論より分かる。

【0278】前記実施例(8)の証明付公開鍵列方法が出力した公開鍵列底と、公開鍵列対と疑似公開鍵列対と、これに付随する応答と挑戦値が検証処理の検証式を満たすことは、

$$\begin{aligned} g'[\mu, 0]^t &= g'[\mu, 0]^{\alpha''} / F_p \\ &= g'[\mu, 0]^{\alpha''} g'[\mu, 0]^{\alpha'} / F_p \\ &= g'[\mu, 1]^c g'[\mu, 2] / F_p \end{aligned}$$

から分かる。

【0279】[健全性]与えられた挑戦値 $c[v]$ ;  $v=1, \dots, n+m'$ に対する、変換情報保有検証処理における検証式を満たす応答 $r[\mu]$ ;  $\mu=1, \dots, n+m$ を求めるには $A[\mu, v]$ ;  $\mu=1, \dots, n+m$ ;  $v=1, \dots, n+m'$ を知らなければならない。

【0280】これは、与えられた  $g[\mu, \Gamma]$ 、 $g'[\mu, \Gamma]$ ;  $\mu=1, \dots, n+m$ ;  $v=1, \dots, n+m'$  に対して、 $A[\mu, v]$ ;  $\mu=1, \dots, n+m$ ;  $v=1, \dots, n+m'$  を知らずして、等価検証処理における検証式を満たす応答を求めることは、離散対数問題を解くことに等しいからである。

【0281】なぜならば、 $A[\mu, v]$ を知らないと言う事は、少なくとも一つの $g'[\mu, \Gamma]$ に関しては、 $g[\mu, \Gamma]$ ;  $\mu=1, \dots, n+m$ を基底とするその表現を知らない。その時、任意の $c$ に関して、検証式を満たす応答を求めることができるなら、 $c[\xi]=1$ 、 $c[v]=0$ ;  $v=0, \dots, \xi-1$ ,  $\xi+1, \dots, n+m'$ なる $c[v]$ を選ぶ事で、離散対数を解くことができるからである。

【0282】また、挑戦値 $c[v]$ は、コミットメント $g[\mu, \Gamma]$ 、 $g'[\mu, \Gamma]$ を引数に持つため、挑戦値が決定してから、コミットメントを調節することができない(挑戦値生成関数がこの性質を持つことを要求する)。そのため、証明者には、挑戦値はコミットメント決定後に与えられた乱数と考えることができる。

【0283】どの $g[\mu, \Gamma]$ の成分に関しても、他の成分を基底とするその表現を知らなければ、検証式を満たす応答を複数作る事は、離散対数問題を解く事に等しい。

$$* v''^{\alpha'} v'^{\alpha''} \prod_{i=1}^n v^{r(i)r(i)} / F_p$$

の $v$ に対する指数部が、

なぜなら、異なる $r[\mu]$ 、 $r'[\mu]$ に関して、検証式が成り立つならば、両辺を互いに割る事により、 $g[\mu, \Gamma]$ を基底とする、非自明な1の表現が得られる。これは離散対数問題を解く事に等しいからである。

【0284】入力文列生成方法により生成される入力文列  $g[\mu, \Gamma]$ ;  $\mu=1, \dots, n+m$ ;  $\Gamma=0, \dots$  は、いずれかの $\Gamma$ に関してなすベクトル  $g[\mu, \Gamma]$ ;  $\mu=1, \dots, n+m$ が、ハッシュ関数で生成されているか、あるいはハッシュ関数で生成されているベクトルを乗じた等の操作で生成されていることが明らかになっているため、互いに、他を基底として表現する事が計算量的に困難である、と考えられる。

【0285】以上により、検証式を満たす $r[\mu]$ ;  $\mu=1, \dots, n+m$ として、 $g'[\mu, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\lambda_{\mu}^{\alpha'}}$  /  $F_p$ ;  $v=1, \dots, n+m'$ なる $A[\mu, v]$ を用いて、 $r[\mu] = \sum_{v=1}^{n+m'} A[\mu, v]c[v] / F_q$ ;  $\mu=1, \dots, n+m$ 、と生成する以外には、証明者は計算できない。個別公開鍵を用いる方法でも同様である。

【0286】上述のようにある $\Gamma$ に関して  $g'[\mu, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\lambda_{\mu}^{\alpha'}}$  /  $F_p$ 、 $v=1, \dots, n+m'$ の関係が証明されたならば、他の $\Gamma$ に関しても以下のように同様に証明される。

【0287】挑戦値生成関数の引数に含めた $g[\mu, \Gamma]$ 、 $g'[\mu, \Gamma]$ に対して検証式が成り立つならば、 $g'[\mu, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\lambda_{\mu}^{\alpha'}}$  /  $F_p$ 、 $v=1, \dots, n+m'$ である。

【0288】なぜならば、 $g'[\mu, \Gamma] = \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\lambda_{\mu}^{\alpha'}}$  /  $F_p$ 、 $v=1, \dots, n+m'$ と表した時に検証式が成り立つならば、 $= \prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\lambda_{\mu}^{\alpha'}} \{ \sum_{v=1}^{n+m'} (A[\mu, v] - A'[\mu, v])c[v] \} = 1$  /  $F_p$ が成り立つ。

【0289】ところが、無作為に選ばれた $c[v]$ に関して、これが成り立つのは、 $\prod_{\mu=1}^{n+m} g[\mu, \Gamma]^{\lambda_{\mu}^{\alpha'}}$  /  $F_p$ 、 $v=1, \dots, n+m'$ の時だけであるからである。



【0290】前記実施例(2)において、変換条件コミットメント生成処理により準元係数をコミットしたもの  $u, u[\mu]; \mu=0, \dots, n$  が与えられたとき、応答  $r[i]; i=1, \dots, n$  と、準応答  $r'$  が検証式を満たす時、準応答  $r'$  は一意であり、

$$r' = \lambda[0] + \sum_{i=1}^n \lambda[i] r[i] r[i] / F_q$$

が、検証式を満たすことより  $r'$  は上式で表されるものである。

【0291】前記実施例(2)の恒等式の検証式の左辺の  $v$  の指数部を展開すると、

$$\begin{aligned} & \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[h, i] A[h, j] A[h, k] c[i] c[j] \\ & c[k] + \sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n (3A[h, 0] A[h, i] A[h, j] + \rho'' \lambda[h] A[h, i] A[h, j]) c[i] c[j] + \sum_{i=1}^n (\sum_{h=1}^n (3A[h, 0] A[h, i] \\ & 0) A[h, i] + 2\rho'' \lambda[h] A[h, 0] A[h, i]) + \rho' A[0, i] c[i] + \sum_{h=1}^n (A[h, 0] A[h, 0] A[h, 0] + \rho'' \lambda[h] A[h, 0] A[h, 0]) + \rho'' \\ & \lambda[0] + \rho' A[0, 0] / F_q \end{aligned}$$

である。

【0292】右辺の  $v$  の指数部は、

$$\sum_{i=1}^n (c[i] c[i] c[i] + \psi[i] c[i] c[i] + \phi[i] c[i]) + \phi[0] / F_q$$

である。

【0293】よって、いかなる  $c[\mu]; \mu=0, \dots, n$  に関しても、検証式が成り立つためには、 $c[\mu] c[v] c[\xi]; \mu, v, \xi=0, \dots, n$  の係数が同じでなければならない。それ以外の時に、検証式が無作為に与えられた  $c[\mu]$  に関して成り立つ可能性は、無視できる。

【0294】これは、

$$\delta[i, j] = 1 \quad i=j \text{ の場合}$$

$$= 0 \quad \text{その他}$$

$$\delta'[i, j, k] = 1 \quad i=j=k \text{ の場合}$$

$$= 0 \quad \text{その他}$$

を用いて、

$$\sum_{h=1}^n A[h, i] A[h, j] A[h, k] = \delta'[i, j, k] / F_q \quad i, j, k = 1, \dots, n$$

$$\sum_{h=1}^n (3A[h, 0] A[h, i] A[h, j] + \rho'' \lambda[h] A[h, i] A[h, j]) = \delta[i, j] \psi[i] / F_q \quad i, j = 1, \dots, n$$

$$\sum_{h=1}^n (3A[h, 0] A[h, 0] A[h, i] + 2\rho'' \lambda[h] A[h, 0] A[h, i]) + \rho' A[0, i] = \phi[i] / F_q \quad i = 1, \dots, n$$

$$\sum_{h=1}^n (A[h, 0] A[h, 0] A[h, 0] + \rho'' \lambda[h] A[h, 0] A[h, 0]) + \rho'' \lambda[0] + \rho' A[0, 0] = \phi[0] / F_q$$

であることを保証する。このことから、 $A[i, j]; i, j=1, \dots, n$  について以下のことが分かる。

【0295】与えられた  $j, k; j \neq k$  に対して、第  $h$  成分が  $A[h, j] A[h, k]$  である  $n$  次元ベクトル  $A[h, j] A[h, k]; h=1, \dots, n$  と、与えられた  $i$  に対して第  $h$  成分が  $A[h, i]$  である  $n$  次元ベクトル  $A[h, i]; h=1, \dots, n$  とを考える。今、 $n$  個のベクトル  $A[h, i]; i=1, \dots, n$  が  $n$  次元空間を張る、すなわち全てのベクトルが  $A[h, i]; i=1, \dots, n$  の線形結合で表せるとする。すると、上式よりベクトル  $A[h, j] A[h, k]; h=1, \dots, n$  は、すべてのベクトル  $A[h, i]$  と内積が0である

ので、

$$A[h, j] A[h, k] = 0 / F_q \quad h=1, \dots, n$$

である。

【0296】このことから、 $n$  個のベクトル  $A[h, i]; h=1, \dots, n; i=1, \dots, n$  のうち、各  $h$  成分が0でないベクトルはひとつしかない。

【0297】また上式より、 $i=j=k$  の時、 $A[h, i] A[h, j] A[h, k] \neq 0$  であるため、ベクトル  $A[h, i]; h=1, \dots, n$  は少なくともひとつは0でない成分を持つ。よって、全てのベクトル  $A[h, i]; h=1, \dots, n$  は、0でない成分をただひとつ持ち、上式より、それは  $1^{1/3}$  である。

【0298】今度は、 $n$  個のベクトル  $A[h, i]; h=1, \dots, n; i=1, \dots, n$  が  $n$  次元空間を張ることを示す。

【0299】ベクトル  $a[h]; h=1, \dots, n$  を、 $n$  個のスカラー  $\kappa[i]; i=1, \dots, n$  を用いて、

$$a[h] = \sum_{i=1}^n \kappa[i] A[h, i] \quad h=1, \dots, n / F_q$$

と表す。

【0300】もし、 $a[h] = 0 / F_q$  ならば、 $\kappa[i] = 0$  であることを示せば、 $n$  個のベクトル  $A[h, i]; h=1, \dots, n; i=1, \dots, n$  が  $n$  次元空間を張ることを示せる。 $a[h] = 0 / F_q$  ならば、上式の両辺に、第  $h$  成分が  $A[h, i] A[h, i]$  である  $n$  次元ベクトル  $A[h, i] A[h, i]$  を掛けると、上式の二つ上の式より、

$$0 = \kappa[i] / F_q \quad i=1, \dots, n$$

となる。以上をもって  $A[i, j]$  が置換行列であるか、置換行列のいくつかの成分に  $1^{1/3}$  を乗じることによって得られる準置換行列であることが示された。

【0301】実施例(1)の恒等式の検証式の左辺の  $v$  の指数部を展開すると、

$$\begin{aligned} & r[0] r[0] + \sum_{i=1}^n r[i] r[i] / F_q \\ & = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A[i, j] A[i, k] c[j] c[k] \\ & + \sum_{j=1}^n (\sum_{i=1}^n 2A[i, 0] A[i, j] + r'[0] A[0, j]) c[j] \\ & + \sum_{i=1}^n A[i, 0] A[i, 0] + r'[0] A[0, 0] / F_q \end{aligned}$$

である。右辺の  $v$  の指数部は、

$$\sum_{i=1}^n (c[i] c[i] + \phi[i] c[i]) + \phi[0] / F_q$$

である。

【0302】よっていかなる  $c[\mu]; \mu=0, \dots, n$  に関しても、検証式が成り立つためには、 $c[\mu] c[v]; \mu, v=0, \dots, n$  の係数が同じでなければならない。それ以外の時に検証式が無作為に与えられた応答に関して成り立つ可能性は無視できる。

【0303】これは

$$\sum_{h=1}^n A[h, i] A[h, j] = \delta[i, j] / F_q$$

$$\phi[i] = \sum_{h=1}^n 2A[h, 0] A[h, i] + r'[0] A[0, i] / F_q$$

$$\phi[0] = \sum_{h=1}^n A[h, 0] A[h, 0] + r'[0] A[0, 0] / F_q$$

であることを保証する。よって、 $A[i, j]; i, j=1, \dots, n$  は正規直交行列でなければ検証式は成り立つ可能性は無視できる。

【0304】前記実施例(3)、実施例(4)においても同様の議論が成り立ち、 $A[i, j]; i, j=1, \dots, n$  は、置換行列か

つ正規直交行列である。そして、これは置換行列であることを意味する。

【0305】【知識隠匿性】再暗号シャッフル証明文において、再暗号シャッフルの情報が計算量的に秘匿されていることを示す。

【0306】再暗号シャッフルの結果 $g^{v'}[v], m^{\mu'}[\mu]$ 以外にも、  
 $r[\mu], r', \phi[i], \psi[i], \omega, v', v'', v, u, u[i], R[\mu], \Phi[i], V, \Omega, v$

等の値が明らかになる。これらは再暗号シャッフルに関する情報を与えている。しかし指数演算された結果でない条件の数より、再暗号シャッフル行列演算に関する未知数のほうが多くなるように、恒等式の係数をコミットして隠蔽すれば離散対数問題をといて条件の数を増やさない限り解けない。ただし変数の数だけでなく未知数の条件の中への現れ方で解ける場合もあるので、若干の調整を行う必要がある。

【0307】

【発明の効果】以上説明したように、本発明によれば、証明付再暗号シャッフルの計算量を、従来の技術と比べて、低減する、という効果を奏する。

【0308】特に、検証処理は事前に計算しておくことができない応用例が多いと考えられるので、検証の計算量を比較すると、入力暗号文の数を $n$ としたとき、従来の技術(1)では、安全変数が160の時冪乗剰余演算が $320n + 2n$ 回必要とされており、従来の技術(2)では冪乗剰余演算が $8(n \log n - n + 1)$ 回必要とされていたのに対して、本発明によれば、冪乗剰余演算は $7n + 14$ 回で済み、 $n > 4$ の時は、いずれの従来の技術よりも冪乗剰余演算が少ない。

【0309】しかも、本発明においては、検証過程で行う冪乗剰余は、個別の冪乗剰余演算ではなく、冪乗剰余演算の積の計算であることから、個別の冪乗剰余演算よりも少ない計算量で計算できるため、更なる高速化が望める、という効果を奏する。

【図面の簡単な説明】

【図1】従来の技術1の構成を示す図である。

【図2】従来の技術2の構成を示す図である。

【図3】本発明の実施例における証明付再暗号シャッフル装置と再暗号シャッフル検証装置との情報の入出力を示す図である。

【図4】本発明の実施例1の証明付再暗号シャッフル装置の詳細を示す図である。

【図5】本発明の実施例1の再暗号シャッフル検証装置の詳細を示す図である。

【図6】本発明の実施例2の証明付再暗号シャッフル装置の詳細を示す図である。

【図7】本発明の実施例2の再暗号シャッフル検証装置の詳細を示す図である。

【図8】本発明の実施例3の証明付再暗号シャッフル装

置の詳細を示す図である。

【図9】本発明の実施例3の再暗号シャッフル検証装置の詳細を示す図である。

【図10】本発明の実施例4の証明付再暗号シャッフル装置の詳細を示す図である。

【図11】本発明の実施例4の再暗号シャッフル検証装置の詳細を示す図である。

【図12】本発明の実施例5の入力文列生成装置の詳細を示す図である。

【図13】本発明の実施例6の入力文列生成方法の情報の入出力を示す図である。

【図14】本発明の実施例6における前処理装置の詳細を示す図である。

【図15】本発明の実施例7の入力文列生成方法の情報の入出力を示す図である。

【図16】本発明の実施例7の入力文列生成方法の情報の入出力を示す図である。

【図17】本発明の実施例7における証明付暗号化装置の詳細を示す図である。

【図18】本発明の実施例7における暗号化検証装置の詳細を示す図である。

【図19】本発明の実施例6と実施例7における証明付公開鍵列生成装置の詳細を示す図である。

【図20】本発明の実施例6と実施例7における公開鍵列検証装置の詳細を示す図である。

【符号の説明】

- 100 入力暗号文列
- 101 暗号シャッフル
- 102 出力暗号文列
- 103 疑似出力暗号文列
- 104 挑戦値生成関数
- 105 挑戦値
- 106 応答
- 200 置換
- 300 入力文列
- 301 入力暗号文列
- 303 公開鍵
- 303 再暗号シャッフル情報
- 304 再暗号シャッフル行列
- 305 再暗号秘密乱数
- 306 情報隠蔽
- 307 シャッフル行列
- 308 元係数
- 309 準元係数
- 310 係数基底
- 311 その他
- 312 証明付き再暗号シャッフル行列装置
- 313 出力暗号文列
- 314 暗号シャッフル証明文
- 315 変換情報保有コミットメント

- 316 変換条件コミットメント
- 317 応答
- 318 準応答
- 319 再暗号シャッフル検証装置
- 400、600、800、1000 入力文列
- 40、601、801、1006 再暗号シャッフル情報
- 402、602、802、1001 再暗号シャッフル行列
- 403、603、803、805、1002、1005 元係数
- 404、604、605、806、1003 係数基底
- 405、602、808、1009 再暗号シャッフル行列作用
- 406、603、809、1010 出力文列
- 407、604、810、1011 出力暗号文列
- 408、605、811、1012 保有コミット
- 409、606、812、816、1013、1022 恒等式係数計算
- 40A、614、823、1017 コミットメント
- 410、607、813、817、1014、1023 恒等式係数
- 411、608、814、818、1015、1024 隠蔽処理
- 412、613、815、822、1016、1028 条件コミット
- 413、615、824、1030、1034 挑戦値生成
- 414、616、825、1031、1035 挑戦値
- 415、617、826、1032、1036 応答生成
- 416、618、827、1033、1037 応答
- 417、810、1011 出力暗号文列
- 418、622、831、1040 再暗号シャッフル証明文
- 419、623、832、1042 変数情報保有コミットメント生成処理
- 420、625、834、1043 変換条件コミットメント生成処理
- 421、624、835、1046、1047 応答生成処理
- 500 挑戦値生成関数
- 501 挑戦値
- 502、706、902、1103、1105 保有検証
- 503 条件検証
- 504、709、906、1110 検証結果
- 505、710 変換情報保有検証処理
- 506、711 変換条件検証処理
- 610、1026 準元係数隠蔽処理
- 612、821、1027 準応答コミット
- 619、828 準応答生成
- 620、829 準応答
- 60A、807、1007 準元係数
- 704、900、1100 挑戦値生成
- 705、901、1101 挑戦値
- 707、903、904、1102 恒等式検証
- 708、905、1107 準応答検証
- 820 準元係数隠蔽処理
- 821 準応答コミット
- 1004 情報隠蔽因子
- 1018 選択処理
- 1019 入力文列
- 1020 保有生成
- 1021 保有コミット
- 1111、1114 条件検証処理
- 1112、1113 保有検証処理
- 1200、1400 基底生成関数
- 1201 入力ベクトル
- 1300、1501、1901 公開鍵列情報
- 1301、1502、1902 秘密鍵
- 1302、1503、1903 疑似秘密鍵
- 1304、1504 証明付き公開鍵列装置
- 1305、1306、1505、1506 分散公開鍵列対
- 1309 前処理装置
- 1308 検証結果
- 1401、1403 公開鍵列基底
- 1402 前処理
- 1404 公開鍵列
- 1500 共通初期値
- 1507 公開鍵列検証装置
- 1508 検証結果
- 150A 共通初期値
- 1600 共通公開鍵
- 1601 個別公開鍵
- 1602 平文
- 1603 暗号化技術
- 1604 秘密乱数
- 1605 疑似秘密乱数
- 1605 証明付き暗号化装置
- 1607 入力暗号文
- 1608 暗号化証明文
- 1609 暗号化検証装置
- 1610、1810 検証結果
- 1700、1701、1702 べき剰余演算
- 1703 暗号文底
- 1704 コミットメント
- 1706、1911 挑戦値生成
- 1707、1912 挑戦値

1 7 0 8、1 9 1 3 応答生成  
1 7 0 9、1 9 1 4 応答  
1 7 1 0、1 9 0 4 基底生成関数  
1 7 1 1 公開鍵基底  
1 7 1 2 平文  
1 7 1 3 暗号文対  
1 8 0 0、2 0 0 0 挑戦値生成関数  
1 8 0 1、2 0 0 3 挑戦値  
1 8 0 2 暗号化検証

1 9 0 0 初期値  
1 9 0 4、2 0 0 0 基底生成関数  
1 9 0 5、2 0 0 2 公開鍵列底  
1 9 0 6 公開鍵列対生成  
1 9 0 7 公開鍵列対  
1 9 0 8 疑似公開鍵列対生成  
1 9 1 5 公開鍵列証明文  
1 9 1 7 公開鍵列対  
2 0 0 4 公開鍵列検証